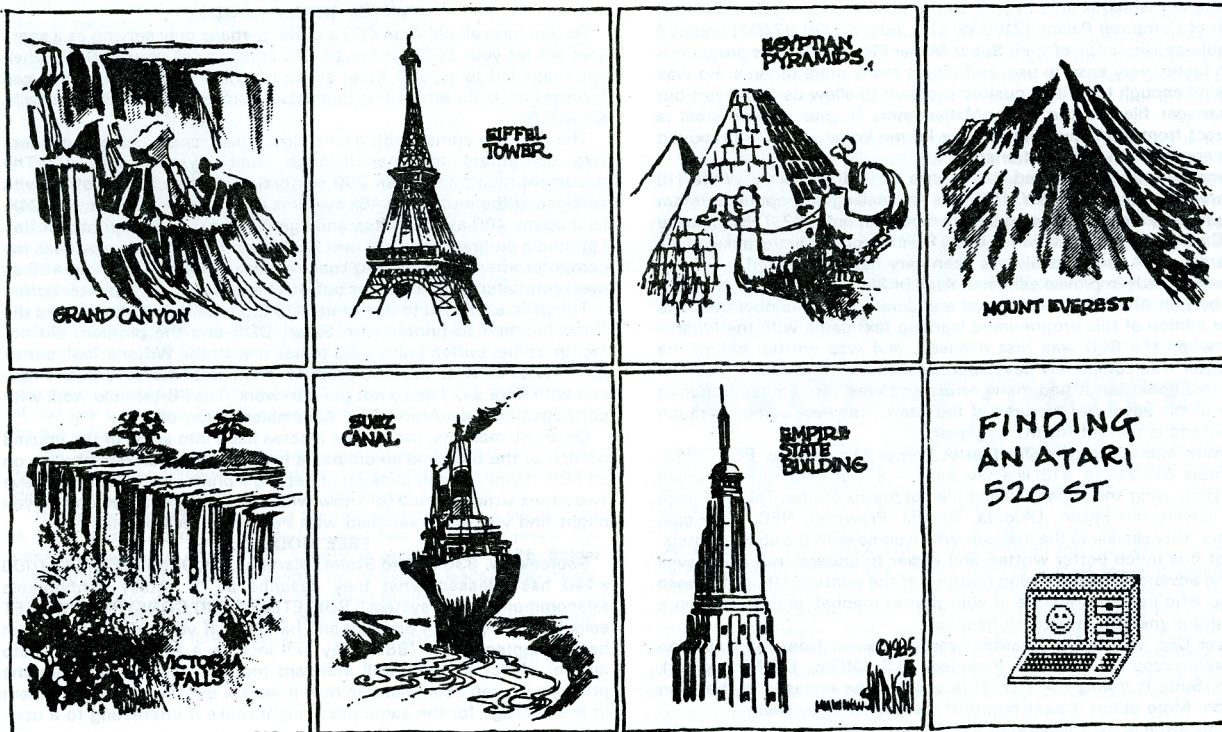


ACE ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

JUNE, 1985

Mike Dunn, Jim Bumpas & Larry Gold, Editors



Eight Wonders of the World

SOMETHING FOR EVERYONE

NEWS AND REVIEWS

by Mike Dunn, Co-Editor

We don't usually talk alot about our meetings of ACE, but the one last month was one of the best, ever! The people at Microbits, led by Kirt Stockwell and their head programmer, demonstrated some of the wonderful capabilities of the new 520ST. We saw amazing graphic demos from Atari, as well as the very high speed of this machine. We understand Sig Hartman from Atari will be delivering a truck load of the new STs when ready to the Portland Atari Club and our club "soon" at a special price.

John Goodrich, a photography teacher at a local high school, demonstrated a truly "awesome" timer/darkroom controller which took him over a year to develop. It connects through the joy stick ports to a "black-box" controller, and controls the various equipment found in a darkroom. The menus, timers and control features were all of very professional quality — as nice as those found in programs by Synapse, etc., and has features way beyond my understanding both in programming and in photography. Hopefully, John will write a series of articles on how he did it.

As you may have noticed, we have been somewhat mixed up with our articles and program listings not appearing in the same issues. Our newsletter is put together by all "slave labor", and many of us have busy professional lives. Different people are responsible for different aspects of the newsletter, and sometimes we don't discover our mistakes and lack of communication until we get the newsletter back from the printers. Sorry, but what do you expect from slave labor...? We do try, even though sometimes that's hard to believe.

Denny and Ruth Hughes have taken care of the very thankless task of the mailing list for the last 2 or so years, but are unable to do so now. We all thank them for all the work, but now it's back in my hands. Since my time is very limited, and since our mailing list needed to be on several files by the program we were using before, we needed a quicker mailing list program which is able to hold all the names on one file. Don Marr, owner of Computer Palace (2160 W. 11th, Eugene, OR 97402) donated a double-density copy of their **Super-Mailer Plus** (\$50). This program is much faster, very easy to use, and stores many more records. He was also kind enough to write a custom program to allow us to convert our Filemanager files to the Super-Mailer ones. If your mailing label is incorrect from the conversion, please let me know — there are bound to be some in this kind of transfer.

Some new items received this month of interest. Jerry White (18 Hickory Lane, Levittown, N.Y. 11756) is now selling his popular program **Poker Tourney** directly to user group members for only \$12. This compiled BASIC program is a simulation of Draw Poker as played in the poker clubs of Gardena, CA, and has always been very highly regarded.

A new, greatly improved edition of **Atari BASIC: XL Edition** (John Wiley, \$15) by Bob Albrecht, Leroy Finkel and Jerald Brown is now out. The earlier edition of this programmed learning text came with the original Atari when the 800 was first released, and was written before the computer was completely developed. Many of us "old timers" learned from this book, but it had many errors and was very limited, ignoring many of the advanced features of the Atari. This new edition is much better, and is perfect for the beginner.

Minute Manual for the Dot Matrix Printer Minuteware, POB 2392, Columbia, MD 21045, \$13) is a new book — a "non-programmers guide to buying, using and understanding the dot matrix printer. This 166 page book covers the Epson, Okidata, Gemini, Prowriter, NEC and Apple printers. Very similar to the manuals which come with the above printers, except it is much better written and easier to understand, and leaves out the advanced programming features of the printers. Of use to those of you who cannot understand your printer manual, and don't have a user group member nearby to help you.

Paper Clip, the highly regarded word-processor from the Canadian software company, Batteries Included, (U.S. Office, 17875 Sky Park North, Suite P, Irvine, CA 92714) is about to be released in the Atari version. More about it next month if we receive it by then.

In this issue is the followup of Stan Ocker's **Labels** program from the March issue. We didn't have room for it in the May issue. Stan also sent a correction to the Label program:

```
88 REC$=INDEX$(128*(S-1)+1,128*S): GOSUB 750: IF S=32 THEN
GOSUB 500: RETURN
89 S=S+1: REC$=INDEX$(128*(S-1)+1,128*S): GOSUB 750: GOSUB
500: RETURN
```

BUMPAS REVIEWS

I can underline! I can print superscripts and subscripts! When I print double columns, the printer backs up the page automatically to the top of the second column after printing the first column! I've never been able to do these functions on my IDS printer with any word processor which uses Atari disk format until now. OSS has sent Version 2.2 of the **WRITER'S TOOL** to all registered owners. This upgrade contains three utility programs to improve the performance of this already great word processor.

The program which allows me to use the printer functions described above is a printer configuration program. It's menu driven — all you have to do is to respond to the prompts for most of the functions. You may also create a translation table to translate characters before they are sent to the printer.

Another program permits the user to customize the default screen formatting parameters. Color and luminance of background and characters may all be set to your taste. Default page length, line spacing, beginning footer line, font, single sheet option, line length, margins, justification and tabs may be set to automatic everytime you boot up the program. Or you may create multiple custom files and "Change" them from within Writer's Tool.

The third big change is the addition of a spelling checker and a dictionary management program. The user can create specialized dictionaries to speed checking. You may add and subtract words from a dictionary already created.

A dictionary is provided on the reverse side of the disk which contains approximately 20,000 words. As the program uses ASCII format, text files created with many other word processors may also be proofed with this checker.

A mail merge function has been added, and one can define up to 10 strings of 255 characters each to be called up each time you press OPTION plus a number from 0-9.

PRINTER BUFFER ROUTINE

Do you have an old Atari 400 around, perhaps only serving as a spare chip set for your 800? Or maybe you couldn't sell your 600 XL when you upgraded to an 800 XL or an XE machine. Well, for \$40 you can buy the Printer Buffer Routine from B.L. Enterprises, Box 4282, Louisville, KY 40204.

The package comes with a cartridge, a disk, and a cable to connect two computers together through their joystick port #1. The documentation (more than 200 sectors) is on the disk, together with versions of the routine for 16k systems and those with more than 24k. I set up my 400 as the buffer, and I got 15.1k to use as a printer buffer. I printed a program listing of over 400 lines. The PBR gave me back my computer after less than 140 lines had been printed! With the 400 as main computer and the 800 as buffer, I have even a larger printer buffer.

The user is advised to use Atari DOS 2.0 with this routine. I tried the Copy function to printer with Smart DOS and the program did not recognize the buffer. I also tried to use it with the Writer's Tool, but as its disk has DOS XL (an OSS product) on it, and I did not create a new one with DOS 2.0, I could not get it to work. The PBR should work with cartridges such as AtariWriter, Assembler-Editor, etc.

On an XL machine, the routine locates itself into some of the unused portion of the 64k, and so creates a larger buffer than one obtains on an 800. If you've been thinking of getting a printer buffer and you have two Atari's sitting around (or know where you can get a 400 cheap) you might find your need satisfied with Printer Buffer Routine.

FREE MODEM!

SourceView, 835 Castro Street, Martinez, CA 94553 800-443-0100 x440 has released what they describe as "a totally configurable telecommunications system." **BULLETIN BOARD CONSTRUCTION SET** sells for \$50, plus \$4 postage and handling. If you order the program before September 1, 1985, they will include a free 300 baud, auto answer, direct connect Bell standard modem. We haven't seen the program, so can't comment on how it works. But the offer of a modem in the package for the same price might make it interesting to a user.

JULY PICNIC

The July picnic will be at Jasper park on the regular meeting day. This year we will have a "flea market". ACE members can bring any surplus hardware and software to sell to other members.

Vice President Larry Gold will designate the area in which members may set up their goods for sale. Each member desiring to sell items will give Larry Gold a \$5.00 donation which will be delivered to WISTEC (Willamette Institute of Science and Technology). WISTEC will send you a receipt for your tax deduction.

SENSITIVE ATARI

(Reprint: SBACE, March, 1985)

The Atari can measure temperature, light level, and humidity. You can add this capability to your computer for just a few dollars, and without making any modifications. The paddle ports of the Atari are actually specialized analog input ports. They can read the value of resistances in the range of 100 to 100,000 ohms with a resolution of one part in 228. The Atari reads the value 60 times a second and puts them in memory locations which can be accessed by Basic Peek statements. This article includes a Simple program which will read the inputs, convert them to useful units of measure, and display them on the screen.

The temperature sensor is a thermistor. Its resistance decreases with temperature. I use a Fenwal model GA45P2 which has a resistance of 50,000 ohms at 25 degrees centigrade. You can use this temperature sensor to measure room temperature, and many other temperatures.

The light sensor is a GE X-6 photocell. Its resistance decreases with increasing light. When the photocell is plugged into the Atari, it can be used in the darkroom to control exposure of prints, or it can be used to measure biological parameters. I've used it to measure the breathing rate of a mouse by shining a focused light beam at the mouse's stomach and pointing the photocell at the edge of the circle of light. When the mouse breathes in, the light reflects into the photocell. When the mouse breathes out, the light reflection moves out of view of the photocell. Similarly, NASA uses photocells to measure the heartbeats of astronauts. They put a small lamp on one side of the ear lobe and the photocell on the back side. Each time the heart beats, the pumped blood darkens the ear lobe enough to change the light level received by the photocell.

The humidity sensor is a Devry Industries Hygropak model HA. It is actually a particle sensor, which is most sensitive to humidity. The higher the humidity, the lower the resistance of the sensor. None of the above sensors cost more than \$5.00.

The program listing demonstrates the capabilities of the Atari analog input system. Lines 10 through 62 Open Device 2 as the keyboard input, put the screen editor in Graphics 2 mode without a text window, and prints an opening menu. Line 75 reads the keyboard. It looks for any one key to be struck. Lines 80 through 90 display the selected key on the screen for a fraction of a second before Line 95 selects the next procedure to execute. If the key is "1", the temperature procedure at Lines 100 to 170 is executed. If a "2", the photocell procedure at lines 200 to 260 is executed. On a "3", the humidity procedure at lines 300 to 360 is executed. If the key is not a 1, 2, or 3, then the closing screen at lines 999 to 1040 is displayed.

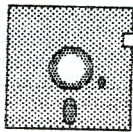
The temperature procedure starts by setting up a Graphics 2 without text window, and turns off the cursor. Line 120 reads the analog input as an 8 bit value between 0 and 228 by doing a Peek (624). Location 624 stores the analog input from the left pot of Jack 1 (there are a total of 8 pot inputs: 2 pots per jack — the shadow locations of these inputs are locations 624 through 631). Line 130 converts the value read to a temperature in fahrenheit. The values for this equation were obtained by exposing the sensor to 2 known temperatures, measuring the input values and using some algebra to solve two simultaneous equations for two unknowns (the slope and intercept of the linear conversion equation). Lines 140 and 150 print the result. Lines 160 and 170 check to see if a key was pressed while the temperature was being read and displayed. If any key was pressed, the value is read and the program goes back to the main menu.

The light sensor and the humidity sensor are handled in a similar fashion. Only the conversion equations are different. Lines 999 to 1040 display the list of sensors demonstrated by the program.

The hardware needed to connect the sensors to the Atari is very simple. A 9-pin female connector is used to plug into the paddle jacks. The connector, which is of the "DB" variety, is available for \$0.75. Solder wires to pins 7 and 9 of the connector and to the two leads of the sensor. Plug the sensor into jack 1, and you are in business.

This demonstration shows the versatility of the Atari analog inputs. The applications are limited only by your imagination.

—Richard Q. Fox



REFORGER '88

(SSI \$59.95) — Every year NATO holds exercises representing a Warsaw Pact attack on West Germany. With REFORGER '88 (SSI, \$60), you can get a piece of the action. This fine wargame simulates an attack on the Frankfurt airbase. You are given command of two corps, with a total of five divisions. Can you fight a delaying action against twelve Warsaw Pact divisions?

This game comes with both one- and two-player options, and there are two scenarios included on the disk. There are also four skill levels, but these only affect the number of replacement units. The game itself comes on a double-sided disk with two plastic covered maps and a detailed rulebook. The map is very high quality Hi-Res graphics, and the text portions of the game have been tabbed over so that there isn't any overscan.

The game format is very nice so it is rather difficult to make a bad mistake. The game components have accounted for every single NATO or Warsaw Pact tank, missile launcher, or armored personnel carrier. The combat system has every weapon rated for attack and defense, with variables for disruption and mode of attack. The actual combat appears a little abstract, and it seems to favor the NATO forces, but it accounts for things which are difficult to accurately measure, such as loss of control, or momentum. An air element is also present, with air squadrons of both sides represented down to the individual planes. There are also four types of weather which only affect air combat.

This is a good game, and I like it, but a few random events thrown in, or perhaps another scenario, might make this a much more enjoyable game. For random events, I suggest having extra or no reinforcements, such as from adjoining units like the 1st Belgian, or the 2nd French. The game also uses the option of chemical warfare. I feel this is not adequately represented in the game. The solitary versions leave much to be desired.

The game stresses a "desperate delaying action", but to me it's more of a case of where the Soviets can be held. With advanced warning, the Soviets can be held to within 10 hexes of their end of the board, while the surprise scenario gives the Soviet armies about half to two-thirds of the board, but still out of reach of their main objective, the Frankfurt airbase. Another criticism is that the game says you must hold until reinforcements arrive. But in actual play, there is only one division for reinforcements. If you can't hold the enemy back with four divisions, you can't hold them back with five.

About the artificial intelligence, it doesn't exist. All I can say is that if the real Soviets fight this bad, NATO has nothing to worry about. The computer is not very flexible, and I have launched localized counterattacks and actually ended up with more units than the Pact forces. Every game I played, a zero score or less is a decisive victory for NATO, I average around -50000. However, with two players, the game is infinitely superior.

As a simulation, the game is not so hot, although as a game system for two players it is excellent. If you have a wargamer for a friend, you'll get many hours of great fun.

— Aaron Ness

MULTIPLE CHOICE

The term "boilerplate" is used by technical writers to describe large portions of text lifted from other handbooks, procedures or other formal documents. Boilerplate needs only to be changed slightly to conform with the new document being written. Not far from a Palimpsest, but neater!! The writer is applying it to portions of computer programs lifted from other computer programs to create a whole new program.

As a result of many trials with school children and the handicapped, I have settled on the Multiple Choice form of quiz as the most useful in presenting quizzes. I tried putting several subjects in one quiz, but the problems involved made those too cumbersome to perpetuate. For a time I made comparisons using type-in answers versus multiple choice answers and I find users will not even try type-in if multiple choice is available. The result of my experience leads me to concentrate on one subject multiple choice programs with simple DATA statements. The Lexicon program embodies the end product of my experience.

The ACE library has the other programs I've prepared in this series, including: French, German, Italian, Portuguese, and Spanish. Lines 100 through 1535 are the same in all these programs except for changes desired for cosmetic purposes. With the exception of Lexicon where the DATA has three words instead of two words, any data will work with any program. In other words, the German introduction and display could be grafted onto the Spanish words and the only thing you'll notice is the German words look like Spanish.

The use of a whole program for one subject has the advantage that more data can be devoted to the subject, the multiple choice answers are more diverse and the READ DATA command executes quickly and without delay, maintaining user interest. This is the only system I have found which allows BASIC to move fast enough to satisfy the user. The teacher can easily make changes to these programs to make them fit the data.

— John R. Kelly

PLATO

(Reprint: P.A.C.E. World, May 1985)

We've all experienced a time when the machines and tools we are using have amazed us. Often it's their power, or their versatility, or sometimes even their stubborn inability to perform the task for which they were designed. Computer users are especially prone to being caught off-guard by their machines. Computers are just about the most complex and easily confused tools we use.

My 4 year old Atari 800 has surprised me time after time; My first taste of Star Raiders; the realization that I, too, could make players and change display lists; an intermittent hardware bug somehow tied to my need to use the computer. My most recent pleasant shock was seeing The Learning Phone (TLP), Atari's Plato access software, in action.

Plato is a telecommunications network operated by Control Data Corp. It supports high resolution graphics and has a truly stupendous library of educational software making effective use of the graphics capability. A unique CDC terminal with special keys and a touch-sensitive screen is normally required to use the system. Unfortunately, the \$10,000 price tag for the terminal restricted Plato use to educational institutions and well-heeled businesses.

Atari microcomputers are among the first to have Plato access software. The **TLP** cartridge was developed for Atari by Vincent Wu and friends, and boy, is it a nice piece of work. They've given Atari computer owners full Plato service with very few of the trade-offs expected when adapting two very different technologies. Simulating the Plato terminal was the hard part. The authors have cleverly used the special abilities of the Atari to take the place of the special functions of the CDC terminal.

The unique Plato keys are replaced by logical combinations of keyboard and Function (Start, Select, and Option) keys. For instance, the Plato "Data" key is Start-D; "Lab" is Start-L, etc. Also, some of the most commonly used Plato keys can be input using joystick, allowing users to "kick back" from the computer and drive the system with their feet on the desk. The joystick can also be used to simulate the touch-sensitive screen on the dedicated CDC terminal. When the system is looking for touch screen input, a cursor appears on the screen. The user moves the cursor to the desired selection, presses the joystick button, and voila, there you are! TLP sends the appropriate signal down the line and the system never knows a touch screen wasn't used.

Two display methods are available at the touch of a button: Normal and Zoom. In Normal, the large 512x512 Plato screen is compressed to fit completely into the Atari Graphics mode 8 (320x192) screen. The Zoom mode restores the full Plato resolution, but naturally only a small portion (about 30%) of the screen can be displayed at one time. The joystick is used to move the smaller Atari "window" over the larger Plato screen, using Atari's fine scrolling ability.

No dry description can take the place of hands-on experience. . . . Getting on Plato is similar to accessing the more familiar information services, like Compuserve and The Source. Plato has no initial fee and presently costs \$7.50/hr. . . .

It checks for Electronic Mail under a handful of categories. It also checks the present participation in a number of multiplayer, interactive games. . . . We then menued up an educational program, one of over 6,000 available on line. It was instructions on how to safely start, operate and slip the Auxiliary Power Unit on a Boeing 747! Plato quickly drew the necessary subsection of the 747 instrument panel, complete with switches and clearly labelled gauges. A small box of text appeared and prompted me to take the first step. At the same time, a bright cursor appeared, indicating I should use the joystick... you can just about get the complete ground school for the 747 with Plato.

Another somewhat randomly selected program was a thorough lesson on the vector relationships of moving objects, complete with animated descriptions and detailed formulas. Plato explains things much better than my college dynamics professor, at a much lower cost! The catalog of educational programs includes programs for elementary students up to graduate students, dairy farmers, electrical technicians, educators, managers, insurance salesmen, etc!

We then called up a list of users presently on the system and made contact with one. Regardless of what they or you are doing on Plato, you may call another user **without interrupting your or their task!** The user we called was playing chess with Plato. He put us in monitor mode and or screen became a duplicate of his, complete with graphics. He showed us some fantastic drawing programs from his files, **files we couldn't have accessed without him!** We carried on a running dialog with him on the bottom two lines of the screen **while we were watching his graphics programs develop on the rest of the screen!**

Next we looked at three of the many realtime, interactive games. . . . Empire is a simulation of tactical space combat, with animated (about 2 second refresh rate) graphics. Wolfpack is a WWII submarine vs destroyer simulation. Dry Gulch is a role-playing simulation of gold-mining in the Old West. Each of these is MUCH too complex to describe in detail. In each the simulation goes on 24 hours/day, as long as there are active players. You enter and leave the simulation at will. While playing you cooperate with your fellow teammates to achieve the common goal, or strike out on your own for individual glory. You can communicate with the other players without interrupting the simulation.

Hopefully, now you've got an idea of how different Plato is from other info services. I've been around computers professionally and as a hobby for years. This is just about the most generally useful service I've seen a computer perform. Atari's TLP brings this to your doorstep.

In all fairness, using Plato with TLP does have some minor drawbacks. Reading the text on the Normal mode screen does take some practice, due to the scrunched characters. Not having the full screen always available in all its exquisite graphic detail is a shame. Things might tend to get expensive using Plato at \$7.50/hour if you're stuck with a 300 baud modem. However, these are not really serious problems when stacked up against the obvious benefit of having this powerful educational tool in your very own home.

Atari should be releasing TLP cartridges before June. At last rumor the packaging was being prepared now. The cost? Hopefully, right around \$30.

— David N. Koster

B/GRAPH

Reichman, Michael H. & Wilson, Robert — 1984

B/Graph has been improved, and is even better than in its original form. It comes to us with one of the best 149 page instruction booklets, I have ever used. The booklet is organized around a series of Tutorials. The nice thing about B/Graph is the software is as impressive as the nicely laid out instructions. Tutorial 1, entitled Graphing, gets you started. This section deals with the basics of start up and utilization of the main menus.

In the booklet each of the main menus is shown by a black and white reproduction of the screen display. A list of subjects shown in the main menu allows you to gain an understanding of the extent and versatility of this program. The main menu lists the following options:

- 1)Reset printer type (allows access to printer options and variables);
- 2)Graphing (allows Bar Graph, 3D Bar Graph, Segmented Bar Graph, Line Graph, Scatter Graphs, Market Graphs, plus grids overlaying rescaling color control);
- 3)Pie Chart (allows the construction of Pie Graphs, like the graph function this also has many customizing features);
- 4)Graph Image/labeling (allows screen dump to a disk as a high resolution picture file, slide display function, alternate labeling);
- 5)File Manipulation (allows file data manipulation, exponential smoothing and moving averages, ASCII files);
- 6)Statistical Analysis (allows distribution testing of data);
- 7)Regression Analysis (allows mathematical curve development for data);
- 8)Correlation Analysis (allows correlation between two factors);
- 9)chi-Square (allows Chi-Square analysis);
- 0)Mini DOS (allows access to basic DOS functions);
- E)Enhancement Disk.

From this brief summary of the main menu I hope you can get a feeling for the range of options open to the users of this program. In addition to a section on each of the above topics, the instruction book contains two appendices and a bibliography. Not being much of a statistician I was happy to see the inclusion of the bibliography. The bibliography gives data on some basic books dealing with statistics and their uses. It also gives the titles of some Atari usage books which will help you understand how to utilize your computer and statistics.

Appendix A deals with the problem of Photographing the screen. The program does support the utilization of printers, but for those who wish to take the full advantage of the color graph options or who are doing slide presentations, this short section gives some helpful tips. Certainly a nice touch.

Appendix B deals with Artifacts, or the concept of color graphics and some of the limitations because of color modes, types of monitors etc. This is also short but again a nice touch.

One feature, which I think is interesting, is the Tutorials give you a variety of files with which to work. The tutorial files range from astronomy, population, Napoleonic Wars, to car defects. This makes me see much more clearly the range of usages of statistics.

Like all programs there are some problems. One problem deals with need to reboot to access certain of the B/Graph functions. Also this program is best utilized using a two drive system. On the whole this program is worth looking into if you need the use of some good statistical analysis, or just need a way to create some nice graph effects.

— Nick Chrones

SYNPRINT

BY

BOB FLOYD

```

10 REM SYNPRINT.BAS by Bob Floyd
   A Print Utility For SynFile+
20 REM Written For Use By SPACE - St.
   Paul Atari Computer Enthusiasts,
   November 25, 1984.
30 REM Printer control codes are
   located in lines 1210 and 1220
   for those with printers
40 REM other than NEC 8023A or Pro-
   writer. See SYNPRINT.IN5 for
   further instructions ( this
50 REM is an ATARIWRITER text file ).
100 POKE 82,2:GRAPHICS 0:DIM D$(768),P
$(512),B$(300),T$(256),H$(256),LL$(15)
,PP$(15),FF$(15),A$(20),S$(8),D$(15),D
E(15)
110 DIM UL0$(3),UL1$(3),ENL0$(3),ENL1$(
3),ENH0$(3),ENH1$(3),CPI10$(3),CPI12$(
3),CPI17$(3),DBL0$(3),DBL1$(3)
120 DIM ITL0$(3),ITL1$(3),UDEF0$(3),UDEF1$(3),L(3),L1(3),L2(3):OPEN #1,4,0,"
K":TRAP 130:GOSUB 1200:GOTO 140
130 TRAP 130: "Error #";PEEK(195);" D
etected at Line ";PEEK(186)+256*PEEK(1
87)
135 ? :? "Press Any Key to Continue":G
ET #1,A
140 CLOSE #2:CLOSE #3:POKE 752,1:GOSUB
250:GET #1,A:IF A>127 THEN A=A-128
150 A=A-48:ON A GOTO 300,500,700,2000,
160:GOTO 140
160 POKE 752,0:END
199 REM PRINT TOP OF SCREEN HEADING
200 ? "K":? " PRINT UTILITY FOR SY
NFILE+":POKE 82,18:
210 ? "Data File: ";FF$:? "Label File
: ";LL$:? "Format File:":PP$
220 POKE 82,2:POSITION 2,5:RETURN
249 REM MAIN MENU
250 GOSUB 200: "MAIN MENU":? :? "1
= Load File":? "2 = Load/Create Label
File"
260 ? "3 = Load/Create Print Format Fi
le":? "4 = Print":? "5 = Quit":? :? "E
nter Choice":RETURN
299 REM LOAD FILE - headings, etc.
300 GOSUB 200: "Load File":? :? :S$="
D:*.TXT":GOSUB 1800:IF M=0 THEN GOSUB
1450:GOTO 140
310 GOSUB 1650:A$(LEN(A$)+1)="TXT":OP
EN #2,4,0,A$:FF$=A$(3,LEN(A$)-4)
320 FOR I=1 TO 4:INPUT #2;H$:NEXT I:IN
PUT #2;T$
330 IF LEN(H$)<LEN(T$) THEN FOR I=LEN(

```

```

H$)+1 TO LEN(T$):H$(I,I)=" ":NEXT I
340 GOSUB 200:I=0:NH=1:D$(1)=1
350 I=I+1:IF I>LEN(T$) THEN 410
360 IF T$(I,I)="-" THEN 350
370 DE(NH)=I-1
380 I=I+1:IF I>LEN(T$) THEN 410
390 IF T$(I,I)="-" THEN 380
400 NH=NH+1:D$(NH)=I:GOTO 350
410 DE(NH)=LEN(T$):? :? "Headings Foun
d":? :GOSUB 1900: "Press Any Key
to Continue"
420 INPUT #2;T$:NOTE #2,SEC,CHAR:GET #
1,A:GOTO 140
499 REM LABEL FORMAT
500 GOSUB 200: "Label Format":? :? :?
"Load or Create":? :S$="D:*.LBL"
520 GET #1,A:GOSUB 1300:IF A=76 THEN 6
30
530 IF A<>67 THEN 140
540 POKE 752,0: "Enter Left Margin
Offset: ";INPUT LM
550 ? :? "Enter # of Columns Assuming"
: " 10cpi: ";INPUT NC
560 ? :? "Enter # of Labels Across Pag
e (1":? " to 3): ";INPUT NL:IF NL<1 O
R NL>3 THEN 580
570 G=0:IF NL>1 THEN ? :? "Enter Horiz
ontal Gap Between Labels":? " Assuming
10cpi: ";INPUT G
580 ? :? "Enter Number of Rows (vertic
ally)":? " of Text per Label: ";INPUT
NR
590 GOSUB 200: "Label Format":? :? :G
OSUB 1800: "GOSUB 1650:A$(LEN(A$)+1)=
".LBL"
610 OPEN #2,8,0,A$:LL$=A$(3,LEN(A$)-4)
620 ? #2;LM: " #2;NC: " #2;G: " #2;NL: " #
2;NR:GOTO 670
630 GOSUB 1800:IF M=0 THEN GOSUB 1450:
GOTO 140
640 GOSUB 1650:A$(LEN(A$)+1)="LBL"
650 OPEN #2,4,0,A$:LL$=A$(3,LEN(A$)-4)
660 INPUT #2;LM:INPUT #2;NC:INPUT #2;G
:INPUT #2;NL:INPUT #2;NR
670 LM=LM:G1=G:NC1=NC:GOTO 140
699 REM PRINT FORMAT
700 GOSUB 200: "Print Format":? :?
710 ? "Load or Create":GET #1,A:CRF=1:
? :S$="D:*.FMT"
720 GOSUB 1300:IF A=76 THEN 1100
725 IF FF$="None" THEN ? "Data Fil
e Not Loaded":GOSUB 1450:GOTO 140
730 IF A<>67 THEN 140

```

```

740 P$="":TRAP 745:GOTO 750
745 TRAP 745: "Error #";PEEK(195);" D
etected at Line ";PEEK(186)+256*PEEK(1
87):GOSUB 1600:GOSUB 1450
750 POKE 752,1:GOSUB 1700: "0 = Strin
g Input":? "1 = Heading":? "2 = Item":
? "3 = Spaces":? "4 = Tab"
760 ? "5 = Remove Trailing Blanks":? "
6 = Remove Leading Blanks-next item"
770 ? "7 = Printer Control Code":? "8
= Carriage Return(s)":? "9 = End Entry
":? :GET #1,A:IF A>127 THEN A=A-128
780 POKE 752,0:A=A-48:ON A GOTO 830,86
0,870,900,910,940,950,1030,1040:IF A<
0 THEN 750
799 REM String input
800 CRF=0:GOSUB 1500: "Enter String:"
:INPUT T$:IF LEN(T$)=0 THEN GOSUB 1400
:GOSUB 1600:GOTO 750
820 P$(LEN(P$)+1)=T$:A=30:GOSUB 1500:G
OTO 750
829 REM heading input
830 CRF=0:GOSUB 1700:GOSUB 1900: "Whi
ch Heading: ";
840 GOSUB 1500:INPUT A:IF A<1 OR A>NH
THEN GOSUB 1400:GOSUB 1600:GOTO 750
850 GOSUB 1500:GOTO 750
859 REM Item input
860 CRF=0:GOSUB 1700:GOSUB 1900: "Whi
ch Item: ";GOTO 840
869 REM Spaces
870 CRF=0:GOSUB 1500: "How Many Space
s: ";INPUT A
880 IF A<1 OR A>255 THEN GOSUB 1400:G
OSUB 1600:GOTO 750
890 GOSUB 1500:GOTO 750
899 REM Tab
900 CRF=0:GOSUB 1500: "Tab to Which C
olumn: ";INPUT A:GOTO 880
909 REM trailing blanks
910 CRF=0:GOSUB 1500: "Leave How Many
Trailing Blanks: ";INPUT A
920 IF A<0 OR A>255 THEN GOSUB 1400:G
OSUB 1600:GOTO 750
930 GOSUB 1500:GOTO 750
939 REM leading blanks
940 GOSUB 1500: "Leave How Many Leadi
ngs Blanks":? " On Next Item: ";INPUT
A:GOTO 920
949 REM printer codes
950 GOSUB 1500:GOSUB 1700: "Printer C
ontrol Codes":?
960 ? " On Off Code":? " 1 2
Underline":? " 3 4 Enlarged"
970 ? " 5 6 Enhanced":? " 7

```


SYNPRINT CON'T

```

8 Double Strike"
980 ? " 9 10 Italics":? " 11 12
    User Defined":?
990 ? " 13 = 10cpi":? " 14 = 12cpi":?
    " 15 = 17cpi"
1000 ? :? "Which Code: ";INPUT A:IF A
>0 AND A<16 THEN 1006
1003 GOSUB 1400:GOSUB 1600:GOTO 750
1006 IF A<3 OR A<4 THEN 1015
1009 IF CRF<1 THEN 1003
1010 IF A>12 AND CRF=0 THEN GOSUB 1400
:GOSUB 1600:GOTO 750
1012 CRF=CRF+1:GOSUB 1500:GOTO 750
1015 IF A<13 THEN 1020
1018 IF CRF<1 THEN 1003
1019 GOSUB 1500:GOTO 750
1020 CRF=0:GOSUB 1500:GOTO 750
1029 REM carriage returns
1030 CRF=1:GOSUB 1500:?"How Many Carr
    iage Returns: ";INPUT A:GOTO 800
1039 REM end entry
1040 IF CRF<1 THEN P$(LEN(P$)+1)="A"

1050 TRAP 130:GOSUB 1700:GOSUB 1800:GO
    SUB 1650:A$(LEN(A$)+1)=",FMT"
1080 OPEN #2,0,0,A$:PP$=A$(3,LEN(A$)-4
):? #2:P$:GOTO 140
1100 GOSUB 1800:IF N=0 THEN GOSUB 1450
:GOTO 140
1120 GOSUB 1650:A$(LEN(A$)+1)=",FMT"
1130 OPEN #2,4,0,A$:PP$=A$(3,LEN(A$)-4
):P$="":INPUT #2:P$:GOTO 140
1199 REM INITIALIZE VARIABLES
1200 FF$="None":LL$=FF$:PP$=FF$
1209 REM NEC or Prowriter Codes
1210 UL0$="EY":UL1$="EX":ENL0$=CHR$(15
):ENL1$=CHR$(14):ENH0$="E":ENH1$="EH"
1220 CPI12$="EE":CPI17$="EQ":DBL0$="":
    DBL1$="":ITL0$="":ITL1$="":UDEF0$="":U
    DEF1$="":
1230 LM=0:NC=100:G=0:NL=1:NR=0:RETURN

1299 REM Strip off inv & lower case
1300 IF A>127 THEN A=A-128
1310 IF A>96 AND A<123 THEN A=A-32
1320 RETURN
1400 POKE 752,1:?"ILLEGAL ENTRY"
:CHR$(253):GOSUB 1450:RETURN
1450 FOR M=1 TO 500:NEXT M:RETURN
1500 P$(LEN(P$)+1)=CHR$(A):RETURN
1550 FOR I=1 TO 3:L(I)=0:L2(I)=0:NEXT
    I:RETURN
1560 A=ENL+1:LM1=INT(0.5+LM*CPI/(10*A)
):G1=INT(0.5+G*CPI/(10*A)):NC1=INT(0.5
+NC*CPI/(10*A)):GOSUB 1550:RETURN
1580 B$=" ":B$(300)=B$:B$(2)=B$:GOSUB
    1550:RETURN
1600 IF LEN(P$)>1 THEN P$=P$(1,LEN(P$)
-1):RETURN
1610 P$="":RETURN
1650 POKE 752,0:?"Enter New Filena
    me: ";INPUT A$:POKE 752,1:GOSUB 1660:
    RETURN
1660 T$=A$:A$(1,2)="D":A$(3)=T$:RETUR
    N
1700 GOSUB 200:?"Create Print F
    ormat":? :RETURN
1799 REM DIRECTORY SEARCH
1800 N=0:TRAP 1820:OPEN #2,6,0,5$:GOTO
    1830
1820 TRAP 130:?"Can't Search Directo
    ry":CHR$(253):GOSUB 1450:RETURN
1830 TRAP 130:?"Disk Inventory":?

1840 INPUT #2:A$:IF LEN(A$)<17 THEN 18
    60
1850 N=N+1:?"A$(3,10):GOTO 1840
1860 CLOSE #2:IF N=0 THEN ? "No Files
    Found"
1870 RETURN
1899 REM PRINT HEADINGS
1900 FOR I=1 TO NH
1910 IF I<10 THEN ? " ";
1920 ? I:?" ";H$(D$(I),DE(I))
1930 NEXT I
1940 RETURN
1999 REM PRINT ROUTINE
2000 IF FF$="None" OR PP$="None"
    THEN ? "Data File or Format File No
    t Loaded":CHR$(253):GOSUB 1450:GOTO 14
    0
2010 POKE 752,1:GOSUB 200:?"Print Ro
    utine":?"Insert Data Disk and Pre
    ss Any Key":GET #1,A
2020 A$="D":A$(3)=FF$:A$(LEN(A$)+1)="
    .TXT":OPEN #2,4,0,A$
2030 POINT #2,SEC,CHAR:BLKFL=0:ENL=0:C
    PI=10:GOSUB 1560:GOSUB 1580
2040 FINFLG=0:?"Specify Output DEV
    ICE:FILENAME ";INPUT A$:IF LEN(A$)=0
    THEN A$="P:"
2043 IF A$="P:" THEN ? "Turn On Pri
    nter & Press Any Key":GET #1,A:?"P
    res START to Pause or Abort"
2046 OPEN #3,8,0,A$:IF A$="P:" THEN ?
    #3:CPI10$:UL0$:ENL0$:ENH0$:DBL0$:ITL0$
    :UDEF0$:
2050 D$=" ":D$(768)=D$:D$(2)=D$:II=0:T
    RAP 2080:IF FINFLG=1 THEN 140
2052 IF PEEK(53279)>6 THEN 2058
2054 ? :?"PRINTING PAUSED - Abort or
    Continue":GET #1,A:GOSUB 1300
2056 IF A=65 THEN 140
2058 FOR I=0 TO NL-1
2060 INPUT #2:T$:IF LEN(T$)=0 THEN 206
    8
2065 D$(I*256+1,I*256+LEN(T$))=T$:II=I
    I+1
2070 NEXT I
2080 NL1=II:TRAP 130:IF NL1=0 THEN 140

2085 IF NL1<NL THEN FINFLG=1
2090 GOSUB 1550:J=0:NR1=0
2100 J=J+1:IF J>LEN(P$) THEN 2133
2110 A=ASC(P$(J,J)):J=J+1:K=ASC(P$(J,J
)):ON A+1 GOSUB 2140,2170,2180,2240,22
    50,2310,2410,2420,2600
2120 IF BLKFL=2 THEN BLKFL=0:GOSUB 270
    0
2125 IF BLKFL=1 THEN BLKFL=2
2130 GOTO 2100
2133 IF NR=0 OR NR1=NR THEN 2050
2136 FOR I=NR1+1 TO NR:?"#3:NEXT I:GOT
    0 2050
2139 REM print string
2140 J1=J
2150 IF ASC(P$(J,J))<30 THEN J=J+1:GO
    TO 2150
2160 T$=P$(J1,J-1):GOSUB 3000:RETURN
2169 REM print heading
2170 T$=H$(D$(K),DE(K)):GOSUB 3000:RET
    URN
2179 REM print item
2180 FOR I=1 TO NL1
2190 N=DE(K):IF DE(K)>LEN(D$) THEN N=L
    EN(D$)
2200 II=(I-1)*256:T$=D$(II+D$(K),II+NL)

2210 GOSUB 3100
2220 NEXT I
2230 RETURN
2239 REM print spaces
2240 T$="":FOR I=1 TO K:T$(I,I)=" ":NE
    XT I
2245 FOR I=1 TO NL1:GOSUB 3000:NEXT I:
    RETURN
2249 REM TAB
2250 FOR I=1 TO NL1
2260 M1=L(I)-L2(I):N=K:IF M>NC1 THEN M
    =NC1
2270 K=(I-1)*100:IF M1<N THEN 2290
2280 FOR II=K+N+1 TO K+M1:B$(II,II)="
    ":NEXT II

```


SYNPRINT CON'T

STUNT RIDER

```

2290 L(I)=N
2300 NEXT I:RETURN
2309 REM remove trailing blanks
2310 FOR I=1 TO NL1
2330 M1=(I-1)*100:TRL=-1
2340 FOR II=L(I)+M1 TO M1+1 STEP -1
2350 TRL=TRL+1:IF B$(II,II)<>" " THEN
2370
2360 NEXT II
2370 IF TRL=K THEN 2400
2380 IF TRL>K THEN L(I)=L(I)-TRL+K:GOT
0 2400
2390 FOR II=M1+L(I)+1 TO M1+L(I)+K-TRL
:B$(II,II)=" ":NEXT II:L(I)=L(I)+K-TRL
2395 IF L(I)>NC1 THEN L(I)=NC1
2400 NEXT I:RETURN
2409 REM remove leading blanks - set
flag & store pointers
2410 FOR I=1 TO NL1:L1(I)=L(I)+1:NEXT
I:BLKFL=1:LEAD=K:RETURN
2419 REM printer codes
2420 ON K GOTO 2430,2440,2450,2460,247
0,2480,2490,2500,2510,2520,2530,2540,2
550,2560,2570
2430 T$=UL1$:GOSUB 3200:RETURN
2440 T$=UL0$:GOSUB 3200:RETURN
2450 ? #3;ENL1$;:ENL=1:GOSUB 1560:RETU
RN
2460 ? #3;ENL0$;:ENL=0:GOSUB 1560:RETU
RN
2470 T$=ENH1$:GOSUB 3200:RETURN
2480 T$=ENH0$:GOSUB 3200:RETURN
2490 T$=DBL1$:GOSUB 3200:RETURN
2500 T$=DBL0$:GOSUB 3200:RETURN
2510 T$=ITL1$:GOSUB 3200:RETURN
2520 T$=ITL0$:GOSUB 3200:RETURN
2530 T$=UDEF1$:GOSUB 3200:RETURN
2540 T$=UDEF0$:GOSUB 3200:RETURN
2550 ? #3;CPI10$;:CPI=10:GOSUB 1560:RE
TURN
2560 ? #3;CPI12$;:CPI=12:GOSUB 1560:RE
TURN
2570 ? #3;CPI17$;:CPI=17:GOSUB 1560:RE
TURN
2599 REM carriage return/print buffer
2600 IF LM1>0 THEN FOR II=1 TO LM1:? #
3;" ":NEXT II
2610 IF NL1=1 AND L(1)=0 THEN 2640
2620 FOR I=1 TO NL1
2625 M1=(I-1)*100:? #3;B$(M1+1,M1+NC1+
L2(I));
2630 IF G1>0 AND I<>NL1 THEN FOR II=1
TO G1:? #3;" ":NEXT II
2635 NEXT I

```

```

2640 ? #3;NR1=NR1+1:IF K=1 THEN 2660
2650 FOR I=2 TO K:? #3;NR1=NR1+1:NEXT
I
2660 GOSUB 1580:RETURN
2699 REM remove leading blanks subr
2700 K=LEAD:FOR I=1 TO NL1
2710 M1=(I-1)*100:TRL=-1:IF L1(I)=L(I
) OR K>L(I)-L1(I) THEN 2850
2720 FOR II=M1+L1(I) TO M1+L(I)
2730 TRL=TRL+1:IF B$(II,II)<>" " THEN
2750
2740 NEXT II
2750 IF TRL=K THEN 2850
2760 IF TRL<K THEN 2810
2780 B$(M1+L1(I)+K,M1+L(I)-TRL+K)=B$(M
1+L1(I)+TRL,M1+L(I))
2790 FOR II=M1+L(I)-TRL+K+1 TO M1+L(I)
:B$(II,II)=" ":NEXT II
2800 GOTO 2850
2810 FOR II=M1+L(I)-K+TRL TO M1+L1(I)+
TRL STEP -1
2820 B$(II+K-TRL,II+K-TRL)=B$(II,II)
2830 NEXT II
2840 FOR II=M1+L1(I)+TRL TO M1+L1(I)+K
-1:B$(II,II)=" ":NEXT II
2850 NEXT I
2860 RETURN
2999 REM general string insertion
3000 IF LEN(T$)=0 THEN RETURN
3010 FOR I=1 TO NL1
3020 GOSUB 3100
3030 NEXT I:RETURN
3100 M1=L(I)-L2(I):IF M1>NC1 THEN RETU
RN
3110 N=LEN(T$):IF M1+N>NC1 THEN N=NC1-
M1
3120 M1=(I-1)*100+L(I):B$(M1+1,M1+N)=T
$(1,N):L(I)=L(I)+N:RETURN
3200 FOR I=1 TO NL1:L2(I)=L2(I)+LEN(T$
):NEXT I
3210 GOSUB 3000:RETURN

```

```

0 REM Thanks to the U.K. Atari Club, P
ob 3, Rayleigh, Essex, England for thi
s program (issue 4)
1 ? "INITIALIZING
":REM "STUNT RIDER" IS A ONE PLAYER,
ONE JOYSTICK GAME.
2 REM PRESS TRIGGER TO START BIKE OFF
THEN INCREASE SPEED OF BIKE BY
PUSHING JOYSTICK UP.
3 REM TO DECREASE SPEED,PULL STICK
BACK.
4 REM THE OBJECT IS TO JUMP THE BUSES
WITHOUT CRASHING.
IF YOU 'OVER DO IT' YOU'LL CRASH.
5 REM **** SEE YOUR BIKES JUMP ****
*** IN SLOW MOTION! ***
6 REM A BONUS OF 200 POINTS IF YOU
'Just' MISS THE LAST BUS.
7 REM POINTS ARE AWARDED FOR OTHER
SUCCESSFUL JUMPS.
YOU HAVE 5 BIKES (OR LIVES).
8 REM SCORING OVER 3000 POINTS GIVES
YOU AN EXTRA BIKE.
9 REM ** PROGRAM WRITTEN BY R.ASKEM **
(FROM IDEA OF YOUNG SON).
89 CEDAR RD.NORHAMPTON.MN1 4RW
10 CB=PEEK(742)*256-1024
14 FOR A=0 TO 511:POKE CB+A,PEEK(57344
+A):NEXT A:FOR A=0 TO 79:READ B:POKE C
B+A,B:NEXT A
15 GRAPHICS 17:POKE 756,CB/256:POKE 70
8,132:POKE 709,52:POKE 710,0:POKE 711,
196
20 POKE 712,200:POSITION 4,4:? #6;"stu
nt rider":POSITION 8,6:? #6;"BY"
25 POSITION 3,8:? #6;"bob and nick":P
OSITION 7,10:? #6;"askew"
30 ST=(PEEK(106)-16)*256:POKE 54279,ST
/256:FOR A=0 TO 256:POKE ST+1024+A,0:N
EXT A:Y=150
35 FOR A=0 TO 7:READ B:POKE ST+1024+Y+
A,B:NEXT A
40 POKE 53277,3:POKE 704,0:SOUND 0,130
,2,2:POKE 559,62
50 FOR X=25 TO 220:POKE 53248,X:NEXT X
:POSITION 4,15:? #6;"HOLD START"
60 IF PEEK(53279)=6 THEN SOUND 0,0,0,0
:POSITION 4,15:? #6;" " "GOTO
850
61 GOTO 50
74 ? #6;"X":POKE 559,0
76 EB=1:U=5:C=0:POKE 712,136
80 POKE 77,0:BUS=INT(RND(0)*11)+6
81 POSITION 8,0:? #6;"SCORE ":C

```

PASSWORD FOR
JUNE IS
SNOW 2-3-4-5

LISTS BY STAN OCKERS

```

10 REM *****
20 REM **      LISTS      **
30 REM **  WORKS WITH DATA FROM **
40 REM **  LABELS PROGRAM 3-85 **
50 REM **    S. O.    MAR 85 **
60 REM **    ACE NEWSLETTER **
70 REM ** 3662 Vine Maple Drive **
80 REM ** Eugene, OR 97405 **
90 REM *****
100 POKE 559,0:GOSUB 5000:GOSUB 2100:G
0SUB 2000:GOSUB 1200:GOSUB 1110:GOSUB
990:POKE 559,34:GOSUB 800:GOSUB 860
101 REM *** FIND LIST LENGTH ***
102 ? "FINDING LAST ENTRY":FOR J=7 TO
4087 STEP 6:IF INDEX$(J,J+4)<>FREX$(1,
5) THEN 106
104 NEXT J
106 LAST=J
110 FOR J=1 TO 4096 STEP 128:COPY$(J,J
+127)=INDEX$(J,J+127):NEXT J
119 REM *** SORT THE INDEX INTO COPY$
***
120 ? CHR$(125):POKE 203,0:POKE 204,4:
POKE 205,6:POKE 206,1: ? "SORTING"
130 A=USR(ADR(SORT$),ADR(COPY$),682)
160 ? CHR$(125):POKE 752,1:GOSUB 500:
GOSUB 400
170 GET #1,A
180 IF A=78 OR A=110 THEN GOSUB 550:GO
TO 160
190 IF A=65 OR A=97 THEN GOSUB 600:GO
TO 110
200 IF A=84 OR A=116 THEN GOSUB 650:GO
TO 110
210 IF A=77 OR A=109 THEN GOSUB 300:GO
TO 160
212 IF A=80 OR A=112 THEN GOSUB 900:GO
TO 160
220 GOTO 170
299 REM *** LIST ENTRIES ***
300 USEFLG=0: ? CHR$(125): ? "WHAT LETT
ER TO START":GET #1,KY
302 ? CHR$(125):SAVKY=KY:FOR K=4087 TO
1 STEP -6:IF ASC(COPY$(K))=KY THEN 3
06
304 NEXT K
306 STPOS=K: ? CHR$(125)
310 POSITION 3,1: ? "ESC-Back to menu,
RETURN, or A-Z":POSITION 3,23: ? "Ar
row keys, Spacebar, ?=Show entry":
320 J=1:POSITION 10,3: ? "1 2 3 4 5
6 7 8":
330 A=ASC(COPY$(K+5)):GOSUB 400:POSITI
ON 3,3+J: ? COPY$(K,K+4)
340 GOSUB 440

```

```

350 J=J+1:K=K-6:IF J=19 THEN 370
360 GOTO 330
370 COL=1:ROW=1:POSITION 8,4: ? CU$:PO
KE 752,0
372 GET #1,KY:IF COL>1 AND KY=43 THEN
GOSUB 420:COL=COL-1:GOSUB 430
374 IF COL<8 AND KY=42 THEN GOSUB 420:
COL=COL+1:GOSUB 430
376 IF ROW>1 AND KY=45 THEN GOSUB 420:
ROW=ROW-1:GOSUB 430
378 IF ROW<18 AND KY=61 THEN GOSUB 420
:ROW=ROW+1:GOSUB 430
380 IF KY>64 AND KY<123 THEN 302
382 IF KY=155 THEN K=STPOS:IF K=108)=L
AST THEN K=K-108:GOTO 306
384 IF KY=63 OR KY=47 THEN GOSUB 450:G
OSUB 850:POSITION 8,2: ? REC$(1,27):GOS
UB 430
386 IF KY=32 THEN USEFLG=1:GOSUB 450:A
=IDX+4097-C*6:B=ASC(PMR$(COL)):C=USR(A
DR(BOL$),A,B):GOSUB 460
388 IF KY=27 THEN 396
390 GOTO 372
396 IF USEFLG=1 THEN GOSUB 800:GOSUB 8
80
398 RETURN
399 REM *** BITS TO ASTERISKS ***
400 STARS$="-----":A=USR(ADR(BIT$
),ADR(STARS$),A):RETURN
419 REM *** CURSOR ROUTINES ***
420 POSITION 8+(COL-1)*3,3+ROW: ? " ";
:RETURN
430 POSITION 8+(COL-1)*3,3+ROW: ? CU$:
RETURN
440 FOR L=1 TO 8:POSITION 7+L*3,3+J: ?
STARS$(L,L):NEXT L:RETURN
450 A=STPOS-(ROW-1)*6:CODE$=COPY$(A,A
+5):GOSUB 740:RETURN
460 A=PEEK(A):P=STPOS-(ROW-1)*6+5:COPI
$(P,P)=CHR$(A):J=ROW:GOSUB 400:GOSUB 4
40:GOSUB 430:RETURN
479 REM *** CHOICES ***
480 POSITION 1,20: ? "
"
482 POSITION 1,21: ? " | Name list
| Modify a list | "
484 POSITION 1,22: ? " | list/list Add o
r Make Print list | "
486 POSITION 1,23: ? "
":RETURN
499 REM *** PRINT LIST NAMES ***
500 S=719:GOSUB 850:POSITION 8,1: ? "AV
AILABLE LISTS":POSITION 5,3: ? "0 - Ev
erything":
510 FOR J=1 TO 8:POSITION 5,2*J+3: ? J;

```

```

" - ":A=(J-1)*16+1: ? REC$(A,A+15):NE
XT J:RETURN
519 REM *** CLEAR PROMPT BOX ***
520 POSITION 2,21: ? BLK$(1,35):POSITI
ON 2,22: ? BLK$(1,35):RETURN
549 REM *** NAME A LIST ***
550 GOSUB 520:POSITION 6,21: ? "List to
name (1-8)?":GET #1,A:IF A=155 THEN
RETURN
552 A=A-48:IF A<1 OR A>8 THEN 550
560 GOSUB 520:LIS$="":POSITION 3,21: ?
"INPUT NAME OF LIST #":A:POSITION 6,22
:INPUT LIS$:L=LEN(LIS$)
570 P=(A-1)*16+1:REC$(P,P+15)=BLK$(1,1
6):REC$(P,P+L-1)=LIS$(1,L):S=719:GOSUB
750
580 RETURN
599 REM *** CLEAR LIST TAGS ***
600 GOSUB 520:POSITION 6,21: ? "Which 1
ist to add to (1-8)?":GET #1,A:A=A-48
:IF A<1 OR A>8 THEN 600
604 GOSUB 520:POSITION 3,21: ? "Which 1
ist to add to list ";A;" (0-8)?":GET
#1,B:B=B-48:IF B<0 OR B>8 THEN 604
610 GOSUB 520:POSITION 3,21: ? "ADD LI
ST ";B;" TO LIST ";A;" (Y/N)?":GET #1
,K:IF K<89 AND K>121 THEN RETURN
620 A=ASC(PMR$(A)):IF B>0 THEN B=ASC(P
MR$(B))
630 FOR J=4091 TO LAST+5 STEP -6:POSITI
ON 10,22: ? J:IF B=0 THEN 640
632 C=USR(QND,PEEK(IDX+J),B):IF C=0 TH
EN 641
640 C=USR(COR,IDX+J,A)
641 NEXT J
642 GOSUB 800:GOSUB 880:RETURN
650 GOSUB 520:POSITION 6,21: ? "List to
take from (1-8)?":GET #1,A:A=A-48:IF
A<1 OR A>8 THEN 650
654 GOSUB 520:POSITION 3,21: ? "List to
take from list ";A;" (0-8)?":GET #1,
B:B=B-48:IF B<0 OR B>8 THEN 654
660 GOSUB 520:POSITION 3,21: ? "TAKE L
IST ";B;" FROM LIST ";A;" (Y/N)?":GET
#1,K:IF K<89 AND K>121 THEN RETURN
670 A=255-ASC(PMR$(A)):IF B>0 THEN B=A
SC(PMR$(B))
680 FOR J=4091 TO LAST+5 STEP -6:POSITI
ON 10,22: ? J:IF B=0 THEN 690
682 C=USR(QND,PEEK(IDX+J),B):IF C=0 TH
EN 691
690 C=USR(MND,IDX+J,A)
691 NEXT J
692 GOSUB 800:GOSUB 880:RETURN

```



```

739 REM *** FIND CODE STRING IN INDEX$
RET. SECTOR '5' ***
740 C=USR(ADR(B$),682,ADR(INDX$),ADR(
CODE$),6,LEN(CODE$)):S=715-C:RETURN
749 REM *** SAVE REC$ TO SECTOR 5 ***
750 IO=USR(ADR(SECRM$),ADR(REC$),5,1):
RETURN
799 REM *** WARNING ***
800 TEMP$=REC$:SAVS=5:S=720:GOSUB 850:
IF REC$(1,10)="zzzzzzzzzz" THEN REC$=T
EMP$:S=SAVS:RETURN
802 REC$=TEMP$:S=SAVS:POSITION 3,10:
"WARNING!!- BE SURE DATA DISK IS IN":P
OSITION 10,12
810 ? "THEN PRESS ANY KEY":POKE 764,25
5
820 IF PEEK(764)=255 THEN 820
822 RETURN
849 REM *** READ SECTOR '5' INTO REC$
***
850 REC$(128)=" ":AREC=ADR(REC$):ASECR
M=ADR(SECRM$):IO=USR(ASECRM,AREC,5,0):
RETURN
859 REM *** READ SECTORS 1-32 INTO IND
EX ***
860 INDEX$=" ":INDEX$(4096)=INDEX$:IND
EX$(2)=INDEX$
870 FOR S=1 TO 32:GOSUB 850:J=(S-1)*12
8+1:INDEX$(J,J+127)=REC$:NEXT S:RETURN
879 REM *** SAVE INDEX$ TO SECTORS 1-3
2 ***
880 POSITION 3,21: "SAVING UPDATED IN
DEX":BLK$
882 FOR S=1 TO 32:J=(S-1)*128+1:REC$=I
NDX$(J,J+127):GOSUB 750:NEXT S:RETURN
899 REM *** PRINTING ***
900 GOSUB 520:POSITION 3,21: "Which 1
ist to print (0-8)?":GET #1,A:SEL=A-4
8:IF SEL<0 OR SEL>8 THEN 900
902 GOSUB 520:POSITION 8,21: "Print [
abels":POSITION 3,22: "Print [Directo
ry listing":
904 GET #1,KY:IF KY=68 OR KY=100 THEN
960
906 IF KY=76 OR KY=108 THEN 910
908 GOTO 902
910 IF SEL>0 THEN SEL=ASC(PMR$(SEL))
920 FOR J=4091 TO LAST+4 STEP -6:IF SE
L=0 THEN 926
924 C=USR(QND,PEEK(COP+J),SEL):IF C=0
THEN 930
926 CODE$=COPY$(J-4,J):GOSUB 740:GOSUB
850:RESTORE 940:MARG=6:GOSUB 1300:FOR

```

```

M=1 TO SKIP: ? #2: "" :NEXT M
930 NEXT J:RETURN
940 DATA 0,1,1,2,255,0,0,3,255,0,0,4,2
55,0,0,5,3,6,255,0,14,7,0,255
959 REM *** SINGLE LINE LIST ***
960 ? #2: "" :IF SEL>0 THEN SEL=ASC(PMR$
(SEL))
962 FOR J=4091 TO LAST+4 STEP -6:IF SE
L=0 THEN 966
964 C=USR(QND,PEEK(COP+J),SEL):IF C=0
THEN 970
966 CODE$=COPY$(J-4,J):GOSUB 740:GOSUB
850:RESTORE 978:MARG=0:GOSUB 1300
970 NEXT J:RETURN
972 REM *** DATA LIST FORMATS PRINTING
***
973 REM *** PAIRS (# BLANKS, FIELD #) *
**
974 REM *** # BLANKS = 255 (CARRIAGE R
ETURN) ***
975 REM *** FIELD # = 255 (END OF DATA
) ***
976 REM *** # BLANKS > 100 (TAB TO COL
. BLANKS-100) ***
978 DATA 0,1,1,2,1,3,160,8,255,0,0,4,1
,5,1,6,3,7,255,0,0,255
980 REM *** TOTAL STRING SEARCH ***
982 REM *** ANALOG #12 P. 84 ***
984 REM *** C=USR(ADR(B$),CMT,ADR(A$),
ADR(DT$)RL,DTL) ***
985 REM *** CNT=RECORD CNT ***
986 REM *** A$=STRING DT$=DESIRED TERM
***
987 REM *** RL=RECORD LEN DTL=DES TERM
LEN ***
990 DIM B$(139):RESTORE 1000:FOR J=1 T
O 139:READ A:B$(J,J)=CHR$(A):NEXT J:RE
TURN
1000 DATA 216,104,104,133,204,104,133,
203,104,133,209,104,133,208,104,133,21
5,104,133,214
1010 DATA 104,104,133,205,104,104,133,
206,169,0,133,212,169,0,133,213,162,0,
160,0
1020 DATA 177,214,224,0,208,2,132,216,
209,208,208,43,232,228,206,240,22,200,
196,205
1030 DATA 240,50,72,152,72,130,160,177
,214,133,207,104,168,104,165,207,24,14
4,219,72
1040 DATA 165,204,133,213,165,203,133,
212,104,162,0,224,0,240,17,224,0,240,6
,160
1050 DATA 0,177,214,162,0,164,216,200,
196,205,208,186,165,208,24,101,205,133

```

```

,208,144
1060 DATA 2,230,209,165,203,208,6,165,
204,240,7,198,204,198,203,24,144,156,9
6,-1
1100 REM * LIVE W/O DOS ANALOG #17 P 5
4 *
1102 REM * IO=USR(ADR(SECRM$),ADR(BUF$
),SECT,FLAG *
1104 REM * FLAG=0 TO READ, 1 TO WRITE
*
1110 DIM SECRM$(44):RESTORE 1120:FOR J
=1 TO 44:READ A:SECRM$(J,J)=CHR$(A):NE
XT J:RETURN
1120 DATA 104,104,141,5,3,104,141,4,3,
104,141,11,3,104,141,10,3,104,104,201
1130 DATA 1,208,7,169,87,141,2,3,208,5
,169,82,141,2,3,169,1,141,1,3,32,83,22
8,96
1199 REM *** USR BOOLEAN FUNCTIONS ***
1200 RESTORE 1210:DIM BOL$(64),PMR$(8)
:FOR J=1 TO 64:READ A:BOL$(J,J)=CHR$(A
):NEXT J:BOL=ADR(BOL$)
1204 RESTORE 1220:FOR J=1 TO 8:READ A:
PMR$(J,J)=CHR$(A):NEXT J:WND=BOL+16:OR
A=BOL+32:QND=BOL+48:RETURN
1210 DATA 104,104,133,204,104,133,203,
160,0,104,104,81,203,145,203,96
1212 DATA 104,104,133,204,104,133,203,
160,0,104,104,49,203,145,203,96
1214 DATA 104,104,133,204,104,133,203,
160,0,104,104,17,203,145,203,96
1216 DATA 104,104,104,133,212,104,104,
37,212,133,212,169,0,133,213,96
1219 REM *** POWER STRING ***
1220 DATA 1,2,4,8,16,32,64,128
1299 REM *** PRINT ACCORDING TO DATA 5
TATEMENT ***
1300 M=1:K=-1:HOLD=0:SKIP=1:COL=1
1302 READ BLNK$,FLDNO:PRT$(M,M)=CHR$(B
LNK$):PRT$(M+1,M+1)=CHR$(FLDNO):IF FLD
NO=255 THEN 1310
1304 M=M+2:GOTO 1302
1310 GOSUB 1394
1320 K=K+2:BLNK$=ASC(PRT$(K)):FLDNO=AS
C(PRT$(K+1))
1322 IF BLNK$=255 AND HOLD=0 THEN ? #2
: "" :GOSUB 1394:GOTO 1320
1330 IF BLNK$=255 AND HOLD=1 THEN SKIP
=SKIP+1:GOTO 1320
1340 IF FLDNO=255 THEN ? #2: "" :RETURN
1350 RESTORE 1400+2*FLDNO:READ POSFLD,
LFLD:L=POSFLD+LFLD-1
1360 IF REC$(L,L)="" " AND L>POSFLD THE

```


LISTS CON'T

```

M L=L-1:GOTO 1360
1368 IF L=POSFLD AND (KY=76 OR KY=100)
  THEN HOLD=1
1370 IF L=POSFLD THEN 1320
1372 IF BLNKS>100 THEN BLNKS=BLNKS-100
:FOR M=1 TO BLNKS-COL:? #2;" "":COL=CO
L+1:NEXT M:GOTO 1390
1380 IF BLNKS>0 THEN FOR M=1 TO BLNKS:
? #2;" "":NEXT M:COL=COL+BLNKS
1390 HOLD=0:? #2;REC$(POSFLD,L);:COL=C
OL+L-POSFLD+1:GOTO 1320
1394 FOR M=1 TO MARG:? #2;" "":NEXT M:
COL=MARG:RETURN
1400 REM *** POSITIONS IN REC$ AND LEN
GTHS OF FIELDS ***
1402 DATA 1,12
1404 DATA 13,15
1406 DATA 28,24
1408 DATA 52,24
1410 DATA 76,16
1412 DATA 92,2
1414 DATA 94,5
1416 DATA 99,12
1418 DATA 111,17
1998 REM *** ML SORT UTILITY COMPUTE!
MAR '82 P. 144 ***
2000 DIM SORT$(126):RESTORE 2010:FOR J
=1 TO 126:READ A:50RT$(J,J)=CHR$(A):ME
XT J:RETURN
2010 DATA 104,104,133,217,104,133,216,
104,133,209,104,133,208,169,0,133,218,
133,207,162,1,165,216,133
2020 DATA 214,165,217,133,215,24,165,2
14,133,212,101,205,133,214,165,215,133
,213,105,0,133,215,164,203
2030 DATA 165,206,240,10,177,214,209,2
12,144,44,240,12,176,19,177,214,209,21
2,144,13,240,2,176,30
2040 DATA 200,196,204,240,227,176,23,1
44,223,169,1,133,218,164,205,136,177,2
14,72,177,212,145,214,104
2050 DATA 145,212,192,0,208,241,232,22
4,0,208,2,230,207,228,208,208,172,165,
209,197,207,208,166,165
2060 DATA 218,201,0,208,144,96
2099 REM *** FORM STRING FROM BITS OF
A ***
2100 DIM BIT$(24):RESTORE 2110:FOR J=
1 TO 24:READ A:BIT$(J,J)=CHR$(A):NEXT
J:RETURN
2110 DATA 104,104,133,204,104,133,203,
160,7,104,104,10,144,6,72,169,42,145,2
03,104,136,16,244,96
5000 DIM TEMP$(128),NAFLD$(20),REC$(12
8),BLK$(50),XSS$(128),A$(1),LIS$(16):L

```

```

FLD=18:? CHR$(125)
5010 DIM COPY$(4096),PRT$(50),INDEX$(4
096),STAR$(8):COP=ADR(COPY$)
5020 IDX=ADR(INDEX$):DIM FREX$(6):FREX
$="ZZZZZ0":DIM CODE$(6):OPEN #1,4,0,"K
":OPEN #2,8,0,"P:"
5030 BLK$="":BLK$(50)=BLK$:BLK$(2)=BL
K$:A=PEEK(16):IF A>128 THEN A=A-128:PO
KE 16,A:POKE 53774,A
5040 COPY$(1)="Q":COPY$(4096)=COPY$:CO
PY$(2)=COPY$:DIM CU$(2):CU$="":RETURN
N

```

STUNT CON'T

```

85 Q=165:J=0:EX=0
100 COLOR 1:PLOT 4,21:DRAWTO BUS,21:CO
LOR 2:PLOT 4,22:DRAWTO BUS,22
105 IF U>1 THEN COLOR 165:PLOT 1,0:DRA
WTO U-1,0
110 COLOR 163:PLOT 2,22:PLOT 3,21
120 COLOR 132:PLOT 0,15:DRAWTO 19,15:P
LOT 0,10:DRAWTO 19,10:PLOT 0,5:DRAWTO
19,5
190 FOR A=0 TO 7:POKE ST+1024+Y+A,0:NE
XT A:X=50:Y=64:RESTORE 1090:FOR A=0 TO
7:READ B:POKE ST+1024+Y+A,B:NEXT A
192 POKE 54206,192:POKE 559,62:POKE 53
248,X:SOUND 0,200,2,2
195 IF STRIG(0)=0 THEN P=1:GOTO 200
196 GOTO 195
200 X=X+P:IF X>220 THEN X=40:GOSUB 250
210 SOUND 0,180-INT(P)*7,2,2
212 POKE 53240,X
215 IF STICK(0)=14 THEN P=P+0.025*EX:E
X=EX+0.125
216 IF STICK(0)=13 AND P>1 THEN P=P-0.
1
240 GOTO 200
250 IF Y>110 THEN POP:POKE 53240,0:M=
0:N=22:GOTO 300
255 RESTORE 1090:FOR A=0 TO 7:POKE ST+
1024+Y+A,0:READ B:POKE ST+1024+Y+40+A,
B:NEXT A:Y=Y+40
260 RETURN
300 K=INT(P)*2
305 IF K<3 THEN K=4
306 IF K>17 THEN K=17

```

STUNT CON'T

```

308 Z=70-(K*4)
309 COLOR 0:PLOT M,N:IF M>0 THEN COLOR
0:PLOT M-1,N
310 FOR A=1 TO Z:NEXT A
320 M=M+1
330 LOCATE M,N,L:IF L=163 THEN Q=166:M
=N-1:COLOR 0:PLOT M-1,N+1
335 LOCATE M,N+1,L:IF L=1 THEN N=N-1:C
OLOR 0:PLOT M-1,N+1:J=1
336 IF J=1 THEN 420
340 GOTO 309
420 COLOR 0:PLOT M,N:COLOR 0:PLOT M-1,
N:SOUND 0,M,10,4
425 IF M=K THEN 460
430 FOR A=1 TO Z:NEXT A:M=M+1:GOTO 420

455 IF N=22 THEN 500
460 COLOR 0:PLOT M,N:COLOR 0:PLOT M-1,
N-1:SOUND 0,M,10,4
462 IF M=19 AND N<22 THEN 700
465 IF M=20 THEN LOCATE M,N+1,L:IF L=1
THEN 800
470 FOR A=1 TO Z:NEXT A:M=M+1:N=N+1:GO
TO 455
500 IF M>BUS+4 THEN COLOR 0:PLOT M-1,N
-1:GOTO 800
501 IF M=BUS+3 THEN C=C+200:BO=1
502 SOUND 0,200,2,8
505 Q=169:COLOR 0:PLOT M-1,N-1:COLOR 0
:PLOT M,N:FOR A=1 TO 40:NEXT A:SOUND 0
,180-INT(P)*7,2,2:Q=165
510 FOR M=M TO 19:COLOR 0:PLOT M,N:COL
OR 0:PLOT M-1,N:FOR A=1 TO Z
520 NEXT A:NEXT M:FOR A=1 TO Z/2:NEXT
A:COLOR 0:PLOT 19,22:SOUND 0,0,0,0
530 C=C+(BUS-3)*10+100
532 POSITION 14,0:? #6;C
535 IF C>HI THEN HI=C
540 IF BO=1 THEN SOUND 1,15,10,6:FOR A
=1 TO 50:NEXT A:POSITION 14,2:? #6;"BO
NUS":SOUND 1,0,0,0:BO=0
550 IF C>3000 AND EB=1 THEN 750
600 IF U=0 THEN 900
610 FOR A=1 TO 500:NEXT A:? #6;"K":GOT
O 80
700 COLOR 0:PLOT M,N
710 FOR A=150 TO 250:SOUND 0,A,2,10:NE
XT A:SOUND 0,0,0,0:U=U-1:GOTO 600
750 U=U+1:COLOR 165:PLOT U-1,0:SOUND 0
,20,10,6:FOR A=1 TO 30:NEXT A:SOUND 0,
0,0,0:EB=0:GOTO 535
800 COLOR 167:PLOT M,N:FOR A=150 TO 24
0:SOUND 0,A,2,8
805 IF A=180 THEN COLOR 168:PLOT M,N
810 NEXT A:SOUND 0,0,0,0:U=U-1:GOTO 60
0

```


USERCOMP

```

9000 REM    'UserComp'  
9001 RAM    USR Compiler  
9002 ROM   Atari Owner's Club 1984  

```

```

9003 CLR :CLOSE #1:GOTO 9250
9004 LINE=LINE-I:GOTO LOOP
9005 FOR LINE=II TO 44:POKE XPOS,Z
9006 OP=USR(ADR(U1$),ADR(K1$),LINE)
9007 GOSUB FILL:PRINT "(#";LINE;") ";
9008 FOR N=Z TO II
9009 PRINT CHR$(PEEK(1536+N));:NEXT N
9010 ERR$=" OP NOT RECOGNISED"
9011 CLASS=USR(ADR(U2$),ADR(OP$))
9012 ON CLASS GOTO 9023,9104,9050,9050
,9020,9026,9040
9013 POKE XPOS,KX:PRINT "E";ERR$;
9014 FOR N=Z TO 96 STEP 8
9015 POKE 712,38:SOUND Z,KX,10,8
9016 POKE 755,3:FOR T=15 TO Z STEP -1
9017 SOUND Z,T,10,T:NEXT T:POKE 755,Z
9018 POKE 755,11:FOR T=T-Z TO KX:NEXT T
9019 NEXT N:POKE XPOS,Z:GOTO LOOP
9023 GOSUB FILL
9024 SOUND Z,KL,10,10:POKE OP+3,13
9025 LO=PEEK(1540):OP=4:GOTO 9088
9026 ON PEEK(1541) GOTO 9030,9128,9031
,9032,9033,9028
9027 REM
9028 G$="5BC":K1=56
9029 GOSUB 9125:GOTO 9010
9030 G$="ADC":K1=24:GOTO 9029
9031 G$="5TA":K1=104:GOTO 9029
9032 K1=Z:OP=32:LO=86:HI=244:GOTO 9088

```

```

9033 K1=234:OP=K1:L0=K1:HI=K1
9034 GOTO 9088
9035 POKE 1540,Z:K1=Z
9036 POKE 1541,3:HI=Z:GOTO 9050
9037 ? F$;:IF PEEK(85)<X1-I THEN 9037
9038 POKE 85,Z:RETURN
9040 DIRECTIVE=PEEK(1541)
9041 ON DIRECTIVE GOTO 9044,9003,9200,
9035,9043
9043 GOTO 9170
9044 POKE XPOS,Z:GOSUB FILL
9045 LINE=LINE-(LINE)I)
9046 L=L-ASC(BYTE$(LINE))
9047 FOR N=I TO 3:POKE OP+N-5,Z
9048 NEXT N:GOTO LOOP
9050 POKE XPOS,9:GOSUB FILL:POKE XPOS,
10
9051 MODE=Z:ANS=Z:HI=Z:BASE=10
9052 M=PEEK(1538):X=(N=88):Y=(N=89)
9053 POKE OP+I,31:FOR N=I TO 8
9054 GET HI,K:IF K=155 THEN K=32

```

```

9055 ERR$=" SYNTAX ERROR ! "
9056 PRINT CHR$(K);:POKE OP+N,K-32
9057 FOR T=I TO LEN(K2$) STEP I+I
9058 M=ASC(K2$(T));J=ASC(K2$(T+I))
9059 IF K=M THEN GOTO (9000+J)
9060 NEXT T:GOTO ERROR
9061 REM
9062 IF BASE=16 OR M)I THEN 9066
9063 IF CLASS<3 THEN GOTO ERROR
9064 SOUND Z,XL,10,10:GOTO 9025:REM 5
9065 BASE=16:NEXT N:REM 5
9066 K=K-7:IF BASE<16 THEN GOTO ERROR
9067 K=K-48:ANS=(ANS*BASE)+K
9068 IF ANS<65535 THEN ERR$=" ADDRESS
9069 POKE OP+N+I,128:NEXT N:GOTO LOOP
9070 IF M)I THEN GOTO ERROR
9071 MODE=8:NEXT N:GOTO LOOP
9072 IF M)I THEN GOTO ERROR
9073 MODE=16+X+Y:NEXT N:GOTO LOOP
9074 MODE=MODE+I:GOTO 9069
9075 MODE=2-X:GOTO 9081
9077 IF MODE=9 THEN 9081
9078 MODE=10:PRINT " ,Y";
9079 POKE OP+N+I,12:POKE OP+N+II,57
9080 REM
9081 ERR$=" MODE UNAVAILABLE!"
9082 HI=INT(ANS/256):LO=ANS-(HI*256)
9083 IF HI>Z THEN MODE=MODE+4
9084 BIT=ASC(BIT$(MODE+I)):POP
9085 OP=USR(ADR(U3$),ADR(U3$),BIT)
9086 SOUND Z,XL,10,10:POKE XPOS,18
9087 IF OP>255 THEN GOTO ERROR
9088 POKE XPOS,19?: "C";S-Z:SOUND Z,X
L,10,10
9089 IF K1>Z THEN J=K1:GOSUB STORE
9090 IF OP<4 AND OP<12 THEN J=OP:GOSUB
STORE
9091 J=LO:GOSUB STORE:K1=Z
9092 J=HI:IF HI>Z THEN GOSUB STORE
9093 ? "+";:POKE XPOS,36:J=INT(L/10)
9094 PRINT CHR$(176+(L/99));CHR$(J+176
-10*(J/9));CHR$(176+L-100*J);
9095 SOUND Z,Z,Z,Z:IF MORE THEN 9131
9096 HI=Z:FLAG=Z:NEXT LINE:GOTO 9200
9097 IF MODE=9 THEN POKE OP+N+I,9:IF Y
THEN MODE=8
9098 IF X THEN MODE=8
9099 GOTO 9081
9100 USER$(L+I)=CHR$(J):L=L+I:S=5+I
9101 BYTE$(LINE)=CHR$(S+FLAG)
9102 PRINT J",";:RETURN
9103 REM
9104 ERR$=" LINE 01-44 ONLY !":ANS=Z
9105 GOSUB FILL:POKE XPOS,9

```

```

9106 POKE OP+I,3:? "E+LINE H";
9107 FOR M=I TO I+I:GET #I,K;K=K-32
9108 IF K=Z OR K=123 THEN 9111
9109 POKE OP+N+I,K;K=K-16:PRINT K;
9110 ANS=ANS*10+K:NEXT M
9111 IF ANS(I OR ANS)44 THEN 9013
9112 IF ANS(LINE THEN 9115
9113 OP=PEEK(1540)
9114 HI=Z:L0=Z:FLAG=126+ANS:GOTO 9088
9115 L0=254:GOSUB 9120
9117 ERR$=" OUT OF RANGE!"
9118 IF L0<128 THEN GOTO ERROR
9119 OP=PEEK(1540):HI=Z:GOTO 9088
9120 FOR N=(LINE-I) TO ANS STEP -I
9121 T=ASC(BYTE$(N))
9122 IF T>128 THEN T=-II
9123 L0=L0-T:NEXT N:RETURN
9125 FOR M=I TO I+II
9126 POKE 1535+M,ASC(G$(M))
9127 NEXT M:RETURN
9128 G$="LDA":GOSUB 9125:POKE XPOS,4
9129 GOSUB FILL:PRINT "DUE LDA";
9130 OP=OP-I:MORE=OP:GOTO 9010
9131 OP=MORE+N:POKE OP,95
9132 MORE=Z:G$="STA":GOSUB 9125
9133 FOR M=I TO 100:NEXT M
9134 POKE 85,5:GOTO 9088
9135 REM
9136 ANS=212:G$="OW"
9137 GOSUB 9145:GOTO 9082
9138 ANS=213:G$="IGH":GOTO 9137
9139 ANS=203:G$="EMP":GOTO 9137
9140 ANS=206:G$="OURCE":GOSUB 9145
9141 GOTO 9154
9142 G$="(DESTN.)":ANS=204:IF PEEK(153
6)=76 THEN G$="(SOURCE)":ANS=206
9143 N=M+8:OP=OP-Z:GOSUB 9145:GOTO 907
8
9145 PRINT G$;:FOR J=I TO LEN(G$)
9146 POKE OP+J+I,ASC(G$(J))-32
9147 NEXT J:RETURN
9150 ANS=106:G$="EMTOP":GOTO 9137
9152 IF BASE=16 THEN 9066
9153 ANS=204:G$="ESTN.":GOSUB 9145
9154 OP=OP+5:? " (+=HI)";
9155 NEXT M
9160 ANS=ANS+1:GOTO 9082
9170 HI=USR(ADR("hhd 07U420 i7,7C1 7U
DEH = TRF DEH,0"))
9172 PAGE=PEEK(HI):POKE HI,PAGE-8
9174 POKE 89,PAGE-8
9176 PRINT " BREAK: 'CONT' TO RETURN"
9177 POSITION I,N23:PRINT "DO NOT INCO
RPORATE LINES AT THIS STAGE!";
9178 POSITION Z,II:PRINT "USR$:USER$

```



```

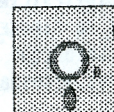
9252 DIM OP$(256),U1$(158),USER$(280),
K1$(26),K2$(64),U2$(64),U3$(64),BYTE$(
100),F$(1),ERR$(24),G$(12),BIT$(N23)
9253 X=USR(ADR("C:\7FNZP\T\KOH\B\"))
M/P/2P/P/P/P/no I/O):POSITION Z,Z
9254 U1$="" No Op. Operand "
9255 PRINT "K";U1$;U1$;" PLA --":
OPEN #I,4,Z,"K":LOOP=9086:XPOS=85
9256 DL=USR(ADR("C:\7FNZP\T\KOH\B\"))
UZO-T/DG-P/H/W/E/P/M-D/H/T/P-P-P-P-
P-M-P-P-E/A/H/W/E/P/M-D/H/T/P-P-P-P-
9257 DL=USR(ADR("C:\7FNZP\T\KOH\B\"))
|||||||AM:N:M:H:V:T:A:I:O:SF:
THRU:1 M/L/2 P/2/G D-m/I")
9258 FOR N=I TO M23-I
9259 POSITION I+(N<10),N:PRINT N;
9260 POKE XPOS,KX+I:PRINT N+22:NEXT N
9261 POKE 752,I:? "U.K ATARI OWNERS C
LUB USERCOMP REV.B":POKE 559,34:F$=
""
9262 OP$="CLD+CLD+CLD+OPR+DEV+INCRIMYH
NOP+PH+CHPLA+PLP+CRIS+SECESEDXTAKESTAYBT5
HTHATYV+ROJ;TKS+PH+ASL+LSRJRO*"
9263 OP$(101)="ADCAND+CMPEORAL+ADORA
SBCESTA+BIT,DEC+CNCND+BP+GDPKXCPYXS
TKESTY+BCCCB+SDDEQ+BNNEPI+IBBL+BOC2"
9264 OP$(197)="GO+RES+BUS+PAUL+ADD+SUB
WDCPKLANSPUT+BYTNCK+BRK+RUB+END"
9265 U1$="(Ch,Ch,K,r,h,e,X,C,h,b,e,H,E%
FZY,DZ,P-i(C,FQUP,P-Z,W,P,R,FY/UPD)
PZ,LH,Q,Q+UKKKH,PZ,i SHPI QV/D)
9266 U1$(183)="MI OFTONID V-V-B-R-KM
/PZFO-COB H+PMKH+POSeP-TKOj,V-U"
9267 K1$="?":*fg9"% %%%▲/()-*-*,I+z"
9268 U2$="(Ch,Ch,N+W-W-%/7/UHL+BJOV/
P/V/BK+BIEN-/)+JJN/HQ:P/QB/Q)
9269 K2$="(AA)BBCCD+EBFBBC1C2C3C4C5C6C
7C8C9C P(FHH-H,M,JYKKA/LHTLS+IME+M"
9270 BIT$="I %%%eet++++++t++++++"
9271 U3$="(H/P/Z/K %h,Ch,Ch,K/H/BHKK
V/h,Ch,KP+KRP+H/C/C/H+CTD+V-U"
9272 USER$="h+":BYTE$="I":GOTO LOOP-I
9290 ? "K COMPILER KAMIKAZE O.K.?"
9291 POKE 712,52:N=9292:INPUT F$
9292 IF F$(0)"Y" THEN 9296
9293 GRAPHICS 0:?:?:? N=N-1
9294 ? "CONT+++":POKE 842,13:STOP
9295 POKE 842,12:IF N)8999 THEN 9293
9296 GRAPHICS 0:LIST :LIST "C":REM DO
NOT USE "SAVE" AS NAME-TABLE IS NOT
CLEARED UNLESS "NEW" IS USED.
```

```

850 FOR T=1 TO 15:READ E: SOUND 2,E,10,
10: SOUND 3,E+1,10,8:FOR A=1 TO 25:NEXT
A:NEXT T: SOUND 2,0,0,0: SOUND 3,0,0,0
860 FOR A=1 TO 18:NEXT A
870 FOR T=1 TO 30:READ E: SOUND 2,E,10,
10: SOUND 3,E+1,10,8:FOR A=1 TO 10:NEXT
A:NEXT T: SOUND 2,0,0,0: SOUND 3,0,0,0
875 FOR A=0 TO 10:READ B:POKE 1536+A,B
:NEXT A:POKE 512,0:POKE 513,6:T=PEEK(5
60)+256*PEEK(561):POKE T+27,134
885 RESTORE 1100
890 GOTO 74
900 POSITION 5,4: ? #6;"game over":POSIT
TION 4,17: ? #6;"HI-SCORE ";HI
905 POSITION 3,12: ? #6;"press trigger"
:POSITION 3,13: ? #6;"to play again"
910 IF STRIG(0)=0 THEN RESTORE 1090:GO
TO 15
915 FOR G=0 TO 6:POKE 708,G
920 NEXT G:GOTO 910
1000 DATA 0,0,0,0,0,0,0,0
1010 DATA 124,254,198,130,130,254,254,
254
1011 DATA 254,66,66,66,126,254,68,68
1020 DATA 1,2,4,8,16,32,64,128
1030 DATA 255,255,0,0,0,0,0,0
1040 DATA 12,2,236,28,90,165,165,66
1050 DATA 26,29,53,82,160,80,80,32
1060 DATA 137,66,32,8,4,28,119,28
1070 DATA 0,0,0,8,4,28,119,28
1080 DATA 0,12,2,236,90,189,165,66
1090 DATA 12,2,236,28,90,165,165,66
1100 DATA 130,0,140,0,89,89,89,0,130,0
,140,0,89,89,89,98,0,105,0,98,0,105,0,
100,0,110,110,0,0,0,90,90,0,0,80,80
1110 DATA 0,0,74,74,74,74,74,74,0
1150 DATA 72,169,196,141,10,212,141,26
,206,104,64

```

SNOW 2-3-4-5
SNOW 2-3-4-5



LOGO BY RUTH ELLSWORTH

```
TO DOGLEFT
  SETSH 7
  FD 2
  SETSH 6
  FD 1
  SETSH 8
  FD 2
  SETSH 6
  JOYSTICK
  END
```

```
TO DOGRIGHT
  SETSH 3
  FD 2
  SETSH 2
  FD 1
  SETSH 4
  FD 2
  SETSH 2
  JOYSTICK
  END
```

```
TO DOGDOWN
  SETSH 1
  FD 5
  JOYSTICK
  END
```

```
TO DOGUP
  SETSH 5
  FD 5
  JOYSTICK
  END
```

```
TO JOYSTICK
  IF ( JOY 0 ) < 0 [STOP]
  SETH 45 * JOY 0
  IF ( JOY 0 ) = 0 [DOGUP]
  IF ( JOY 0 ) = 4 [DOGDOWN]
  IF ( JOY 0 ) = 1 [DOGRIGHT]
  IF ( JOY 0 ) = 2 [DOGRIGHT]
  IF ( JOY 0 ) = 3 [DOGRIGHT]
  IF ( JOY 0 ) = 5 [DOGLEFT]
  IF ( JOY 0 ) = 6 [DOGLEFT]
  IF ( JOY 0 ) = 7 [DOGLEFT]
  END
```

```
TO BEGIN
  LOADDOGS
  PU
  SETSH 1
  WHEN 15 [JOYSTICK]
  END
```

TO GO

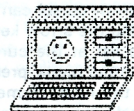
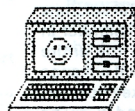
```
PU
RT 90
SETSH 3
FD 15
WAIT 10
SETSH 4
FD 15
WAIT 10
SETSH 3
FD 15
WAIT 10
SETSH 4
FD 15
WAIT 10
LT 90
MAKE "CHAR RC
IF :CHAR = "G [GO]
IF :CHAR = "S [START]
END
```

```
TO TURN
  SETSH 2
  WAIT 10
  GO
  END
```

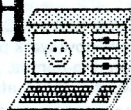
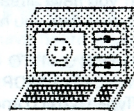
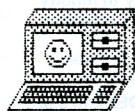
```
TO START
  LOADDOGS
  HT
  PU
  SETSH 1
  ST
  MAKE "CHAR RC
  IF :CHAR = "G [TURN]
  END
```

```
TO LOADDOGS
  PUTSH 1 [127 127 73 85 28 28 8 28 62 6
    2 62 62 62 62 119 119]
  PUTSH 2 [12 28 24 30 28 158 140 156 19
    0 254 254 254 254 126 115 59]
  PUTSH 3 [12 28 24 30 28 30 140 156 190
    254 254 123 115 96 48 48]
  PUTSH 4 [12 28 24 158 156 142 236 252
    254 126 126 110 54 50 3 3]
  PUTSH 5 [127 127 93 93 28 28 8 28 62 6
    2 62 62 62 62 119 119]
  PUTSH 6 [48 56 24 120 56 120 57 49 121
    125 127 127 127 126 206 220]
  PUTSH 7 [56 56 24 120 56 121 49 57 125
    127 127 222 206 6 12 12]
  PUTSH 8 [48 56 24 120 57 121 49 55 127
    127 126 126 118 108 204 192]
  END
```

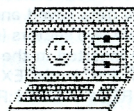
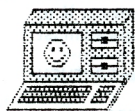
```
MAKE "CHAR "S
MAKE "DOG [127 127 73 85 28 28 8 28 62
  62 62 62 62 62 119 119]
MAKE "DOG2 [12 28 24 30 28 158 140 156
  190 254 254 254 254 126 115 59]
MAKE "DOG3 [127 127 73 85 28 28 8 28 6
  2 62 62 62 62 62 119 119]
MAKE "DOG4 [12 28 24 30 156 158 140 23
  6 254 254 126 126 110 54 51 3]
```



MEETING
WED
JUNE 12TH
SOUTH
EUGENE
HIGH



7:30PM



PASSWORD IS
(SNOW 2-3-4-5)

LISTS

Lists is a program to be used with data created on a data disk by the 'Labels' program in the March '85 issue of A.C.E. Newsletter. It allows you to create up to eight individual lists. These lists are given separate names and can be combined or taken away from each other to create new lists. Any list can then be printed on labels or as a directory.

After a little time for initialization, you will be presented with the current list names and a menu. 'N' allows you to rename a list while 'A' or 'T' permit adding to or taking away from a list. List #0 is made up of all entries, so adding list 0 to any other will produce another list containing all entries. Taking list 0 away from any other will completely clear out that list. Other combinations can be made by adding or subtracting one list from another.

The 'M' option allows you to examine and change the members of any list (except 0). You will be asked for a letter where you want to start displaying members. A list of 18 code keys will be shown with 8 columns indicating which lists contain these members (marked with asterisks). A arrow cursor can be run up and down the rows and across the columns using the arrow keys. Pressing the spacebar will toggle the member in the list at the current cursor location. If you want to see the entire member name, press the question mark (?). 'RETURN' will advance to the next 18 members. You may also press any letter key to start displaying members with that letter. 'ESC' causes the index to be updated and returns you to the menu.

'Print a list' option asks for a list # and asks whether you want labels or a directory printed. Labels are the single width 3 1/2 X 15/16 inch size with tractor feed backing. Be sure to position at the top line of the 1st label before starting.

Unfortunately some changes to the 'Labels' program need to be made to produce a data disk to work properly with the sorting procedure in this program. Instead of X's we need z's (small letter) written in sectors 1-32 and 720. The necessary changes in 'Labels' are:

- 1) In line 5010: XSS\$="z"
- 2) In line 5020: FREX\$="zzzzzz"

If you have already typed in a bunch of names, you can try to modify the data disk you have: Run 'Labels' and hit RESET; Type in the following lines:

```
0 FOR J=1 TO 4096:IF INDEX$(J,J)="X" THEN INDEX$(J,J)="z"
1 NEXT J:STOP
```

Type GOTO 0 and when the process stops you can list INDEX\$ to see that the changes have been made. If they have then type S=720:GOSUB 880 to update the index.

You also need to fill sector 720 with z's by typing in the following in immediate mode:

```
REC$(1)="z":REC$(128)=REC$:REC$(2)=REC$:S=720:GOSUB 750.
```

The data disk should now work properly with the 'Lists' program. If you haven't started entering names, just create a data disk by going to the subroutine at 830 of 'Labels' after making the changes above.

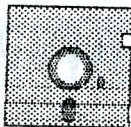
***** WARNING *****

There is a major bug in the 'Labels' program. It has to do with the situation when a 6 byte index code key crosses one of the 128 byte sector boundaries. Since only one sector is saved, only part of this index key gets saved and you can't locate the entry again. The answer is to save two sectors (unless it is the last one, 32). The Following lines must be inserted in the program:

```
77 REC$=INDEX$(128*(S-1)+1,128*S):GOSUB 750:IF S=32 THEN
REC$=TEMP$:RETURN
78 S=S+1:REC$=INDEX$(128*(S-1)+1,128*S):GOSUB 750:REC$=
TEMP$:RETURN
88 REC$=IND-9b-EX$(128*(S-1)+1,128*S):GOSUB 750:IF S=32 THEN
GOSUB 500:RETURN
89 S=S+1:REC$=INDEX$(128*(S-1)+1,128*S):GOSUB 750:GOSUB
500:RETURN
```

I apologize for any inconvenience this may have caused.

Stan Ockers



SYNPRINT

(A Printing Utility For SynFile+ Written By Bob Floyd
For Use By SPACE—St. Paul ATARI Computer Enthusiasts)

This program was written to overcome some limitations in printing labels on SynFile+. You can now print labels 1, 2, or 3 across easily and save your print format to disk for later use. You can also save your label formats for later use. This makes it easy to choose any label format to go with any print format. You must have SynFile+ in order to make use of this program, although I'm sure modifications are possible for use with other programs.

GETTING STARTED

SynPrint contains no search capability. You are expected to let SynFile+ do the searching. The first thing to do is use the "LIST" option of the "REPORTS" function of SynFile+. The best way to do this is to type out all of your headings in the order shown on the screen — consistency is important. Always do this step the same way every time! Next choose the disk output option instead of the printer. Choose 999 as the printed page length. The title will not matter. Then perform any searches that are necessary. It is probably simplest to save your file to the same name used by SynFile+. It will receive ".TXT" extender automatically.

USING SYNPRINT

Load SYNPRINT.BAS the way you would any BASIC program. Probably your first choice will be "1", to load your file. A directory will appear with all files with a ".TXT" extender. Just type in the correct one. SynPrint scans the file and extracts the headings to let you know its found everything alright.

Then you can choose option "2", to load or create a label file. This file tells the print routine how to layout your labels. Left margin offset, # of columns, # of labels per row, # of lines per label, etc. are supported. You should save a label format file for each type of label you use. A disk inventory routine makes saving and loading easier.

Option "3" is for loading or creating print format files. This routine is similar to that used in SOFTSIDE MAGAZINE's database. However, this is a souped up version with many new capabilities. Basically, you must choose from a list of options how to lay out the label. Generally, you can choose any combination of the following items:

String Input—Enter a string from the keyboard to be used in output. It may contain printer control codes, but this may confuse output for multiple labels across.

Heading—The heading (field name) of your choice will be printed.

Item—The item (field data) of your choice will be printed.

Spaces—Output the desired number of spaces.

Tab—Use this command to tab to the desired column number on the label.

Trailing Blanks—Remove undesired trailing blanks and choose how many can be retained. This is handy for printing city, state, etc. Use string input for the comma between city and state. This option can also add spaces if necessary.

Leading Blanks—Remove undesired leading blanks on next item. Again, a chosen number of leading blanks can be left. If a large number of leading blanks are chosen, some of the characters may be lost off the end of the item. The "leading blanks" option is useful for right justified fields (lookup tables are right justified).

Printer Codes—You can choose from the following printer codes assuming your printer provides these capabilities: underlined, enlarged, double strike, italics, user defined, and 10, 12 or 17 characters per inch (i.e., font width). The only rules are that font width changes, if used, must be the first command for a given line. This is also true for enlarged text on/off commands. If both are used, the font width must be first. The program prevents incorrect entries. SynPrint is currently set up for NEC 8023A/C and Prowriter control codes. These codes can be easily changed for various printers by altering lines 1210 to 1220 of the program. The variables should be self explanatory. It is best to set unuseable codes to equal "", a null string.

It is best to plan out your commands for the print format routine. It is easy to make a mistake, so a written copy of your commands helps to get it right the next try. A disk inventory helps in saving and loading your print format files.

Option 4 is for printing. The ability to pause or abort is built into the routine.

CONCLUSION

SynPrint is reasonably fast, but not like lightning. Care has been taken to keep all math as integers so it could be compiled. I have not tried this yet, but expect it to work. I prefer the uncompiled speed for ease in pausing and aborted. Also, if you get into trouble and hit the "break" key, you can re-enter the program safely by typing "GOTO 140". If you have any questions or embellishments, you can contact me at 1193 Churchill St., St. Paul, MN 55103, (612)-487-2627.

RAMTALKER UPDATE

(Reprint: STATUS, Feb. 1985)

(This improvement permits XL and XE users to use RamTalker. — Ed.)

It's time to improve RamTalker. We formerly had an elementary and wrong plotting routine for the sound waveforms. The routine did not plot Frequency vs. Time as I stated, but rather Time vs. Amplitude and Frequency. Trying to plot three values on two axes doesn't always work, so this month's program addition will be a new plotting routine in three dimensions.

The routine asks you which section of sound you want to see. Since the 3D plot takes up more room on the screen, it is not possible to put all 7 seconds worth of sound on the screen at one time. After you tell the program which section to display, it begins to fill the three axis graph on the screen. Each line on the time axis represents 1/20 of a second; the height of the peak represents how many times (frequency) the volume level (amplitude 0-255, top of screen to bottom) occurs at the instant of time.

To place the new routine into RamTalker, follow these steps:

1. Type in the listing. This listing also changes the port to use to Port 2, so everyone can now enjoy RamTalker;
2. After typing the lines, LIST them to disk or cassette as, LIST"D:FIX" or LIST"C:";
3. Load in the earlier RamTalker;
4. Delete these lines: 610, 620, 630, 640, 650, 660, 670, 680, 710, 720, 730, 740, 750, 760, 770, 780, 790;
5. ENTER the new listing from disk or cassette;
6. Now SAVE the newly merged program to disk or cassette.

— Randy Holmes

RGB UPDATE

The May, 1985 issue of the Bay Area Atari User Group Newsletter contains information from Arthur Blomseth, 16987 Frank Ave., Los Gatos, CA 95051. He had to modify the RGB conversion to use his IBM compatible RGB monitor with the Atari.

He added a pot in the SYNC circuit, changed the sync transistor from a 2N2222 to a 2N2907 and referenced all output lines to ground.

ERACE DISK No. 9

SPANISH1, SPANISH2, ITALIAN1 and ITALIAN2 are Four language programs. 2 are in Spanish/English and the other two are in Italian/English. Both programs allows translation both ways. Spanish and Italian 1 are multiple choice where as Spanish and Italian 2 answers' are typed in.

WORLDCAPI & CAPWORLD: These geographical tests have 161 independent nations to tease your memory. There are Direct Answers (CAPWORLD) and Multiple Choice (WORLDCAPI).

ASTROMATH, From the Best of ACE #10. Instead of finding the final product, you find the missing number. Uses the joystick & trigger to shoot down the correct number.

ERACE DISK No. 10

STARBIRD, From Paul Freeman. This program lets you improve your typing skills while having fun.

CRICKET, From England's Page 6 by Phil Griffen. This unusual math game plays a game of England's baseball while answering question of adding, subtract and multiplication.

MAGICBOARD, A multiplication game. Instruction are included. Messages occur on the blackboard telling if THE answer is correct or wrong.

UNSCRMB1, UNSCRMB2, & UNSCRMB3. Three games from John Kelley. The object is to unscramble the letters to spell the word correctly.

SPELLBOUND By Sydney H. Brown. An interesting version of hangman. It plays as a tug of war with a rope between the right- and wrong- lettered teams.

FORKLIFT By Stan Ockers. Take 3 rhyming words from a pile and stack them in a pile on the right.

FLASHO By Jim A. Carr. The object of Flasho is to help memorize the multiplication facts. The name Flasho comes from the flashing 'O' that is positioned by the joystick to select the answer.

— Nora F. Young
ERACE SIG Editor

ACE LIBRARY

Two disks which may be of interest to Basic Programers and Koala Pad artists contain excellent Utilities written by Bob Floyd and Paul Freeman respectively.

Bob Floyd's Disk is a program "Title" program enabling the user to easily add a MicroPainter/Koala screen and supplement it with Scrolling Text, Large text, Color Changes, and "the Atari Rainbow" in the user's choice of color register.

This program, entitled THE BEAR ESSENTIALS, comes with full documentation on disk as well as routines to incorporate the programs into the user's program choice of FORTH and/or BASIC.

The second disk is full of routines by Paul Freeman. It includes some special Koala loaders and Picture Dumps as well as a Utility to superimpose an unlimited number of Koala pictures.

Also on this disk is a special Title generating program for the Basic Programmer . . . and a **great** program for showing Koala/Micropainter Pictures containing user adjustable parameters. This latter program is the best all around "slide show" program I have used.

Both of these disks are excellent and worthy additions to the ACE Library.

— Graham Smith

VP RAMBLINGS

As I write this there still aren't any STs. We have heard from a lot of people as to when they will be out, but so far all of the predictions have proven to be false. I think we will just have to wait until the powers that be decide to get them to market and in what form. So hang in there all you people who want one your day is coming.

If you have gone down to your local software store you have probably found out there isn't much new in software for the Atari. It seems most of the software people are waiting for the STs. This is too bad because they are forgetting about all us users with plain old 400, 800, XL and XE machines. This is a big market which hasn't been used up and still will buy software if it is out there. My next thought is that if you were to write to some of the software companies and tell them you are waiting maybe they will get it in their heads that we do want new products and they just might start to produce again. Maybe?

If some of you have nothing better to do you might also write me and let me know what you think are the 10 best programs you own and why. If you are a little lazy just tell the the 10 best. I want to know what you think are the best programs and also what you the reader likes so I can write about programs which are of interest to you and not just what comes along or is just of interest to me.

— LARRY GOLD

SNOW 2-3-4-5

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to Atari Corp. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$14 a year for U.S., and \$24 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.

Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

****President—** Dick Barkley, 2907 Wingate, Eugene, OR 97405
503-344-5843

Vice Pres—Larry Gold, 1927 McLean Blvd., Eugene, Or 97405
503-686-1490

Librarians—Ron and Aaron Ness, 374 Blackfoot, Eugene, Or 97404
(503)689-7106.

Editors— Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405
503-344-6193

Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405
503-484-9925

E.R.A.C.E. (Education SIG Editor)—Nora Young, 105 Hansen Lane, Eugene, Or 97404 /503-688-1458

Send 50c stamps or coin (\$1 overseas) to the Ness' for the new, updated ACE Library List-new in May 84 !

Bulletin Board

(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives, an Epson MX80 printer, a MicroBits 1100E Modem; running the FoReM Bulletin Board software distributed by MicroBits.

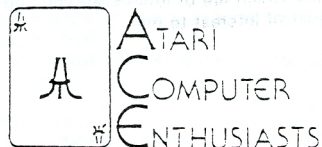
Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only \$12 each (\$2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kaneohe, HI 96744



3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL

TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

SortFinder 1.2

A composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only \$6 for ACE member from:

Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333